Splitting Methods for Efficient Combinatorial Counting and Rare-Event Probability Estimation

Zdravko Botev, BSc(Honours)

A working paper (technical report) written at the University of Queensland May 2009

School of Mathematics and Physics

Abstract

This working paper is divided into two major parts.

In the first part we describe a new Monte Carlo algorithm for the consistent and unbiased estimation of multidimensional integrals and the efficient sampling from multidimensional densities. The algorithm is inspired by the classical splitting method and can be applied to general static simulation models. We provide examples from rare-event probability estimation, counting, optimization, and sampling, demonstrating that the proposed method can outperform existing Markov chain sampling methods in terms of convergence speed and accuracy.

In the second part we present a new adaptive kernel density estimator based on linear diffusion processes. The proposed estimator builds on existing ideas for adaptive smoothing by incorporating information from a pilot density estimate. In addition, we propose a new plug-in bandwidth selection method that is free from the arbitrary normal reference rules used by existing methods. We present simulation examples in which the proposed approach outperforms existing methods in terms of accuracy and reliability.

Keywords:
splitting method, MCMC, rare-event probability estimation, convergence diagnostic, sequential importance sampling, Boolean Satisfiability problem, kernel density estimation, automatic bandwidth selection

Australian and New Zealand Standard Research Classifications (ANZSRC):
010405,010401,010303,010406

Contents

List of Tables vi
List of Figures vii

I The Generalized Splitting Method for Combinatorial Counting and Rare-Event Probability Estimation 1

1 Introduction to Part One 3

2 Classical Splitting for Dynamic Simulation 7

3 Generalized Splitting Algorithms 11
3.1 Generalized Splitting Method 11
3.2 An Adaptive Generalized Splitting Algorithm 21
3.3 Fixed Effort Generalized Splitting 23

4 Extensions of the Splitting method 31
4.1 Estimation of Integrals of the Form $\mathbb{E}_p[H(\mathbf{Z})]$ 31
4.2 Combinatorial Optimization via the ADAM 38
4.3 Sensitivity Analysis 46
4.4 MCMC Sampling 49
4.5 Improved Importance Sampling 62

5 Splitting Methods Synopsis 65

II Kernel Density Estimation via Diffusion 67

6 Introduction to part Two 69

7 Density Estimation with a Gaussian kernel 73

8 The Diffusion Kernel Density Estimator **79**
8.1 The Diffusion Estimator 79
8.2 Bias and Variance Analysis 83
8.3 Special Cases of the Diffusion Estimator 85

9 Bandwidth Selection Algorithm **87**
9.1 Bandwidth Selection for Gaussian Kernel Estimator 87
9.2 Bandwidth Selection in Higher Dimensions 92
9.3 Bandwidth Selection for the Diffusion Estimator 95
9.4 Numerical Experiments 98

10 Diffusion Estimator Synposis **103**

A Appendix **105**
A.1 Gaussian kernel density estimator properties 105
A.2 Proof of Lemma 8.2.1 106
A.3 Proof of Theorem 8.2.1 112
A.4 Consistency at Boundary 113

References **115**

List of Tables

3.1 The sequence of levels and splitting factors used in Algorithm 3.1.1 for instance `uf20-01`. 18

3.2 The sequence of levels and splitting factors used in Algorithm 3.1.1 for instance `uf75-01`. 20

3.3 The sequence of levels and splitting factors used in Algorithm 3.1.1 for instance `RTI_k3_n100_m429_0`. The sequence was generated using the ADAM algorithm with $N = 10^3$ and $\varrho = 0.5$. 28

3.4 Comparative advantages and disadvantages of the GS, FE-GS and ADAM algorithms. 29

4.1 Levels and splitting factors used to compute the normalizing constant of $h(\mathbf{z})$. . 34

4.2 Typical evolution of Algorithm 3.2.1 as an optimization routine for knapsack problem. The maximum value for this problem is 6704. 40

4.3 Case studies for the symmetric TSP. The experiments were repeated ten times and the average minimum and maximum of $S(\mathbf{x})$ were recorded. The parameters of the ADAM algorithm are $\varrho = 0.5$, $N = 10^2$ and the 2-opt updating is repeated $b = n \times 50$ times, where n is the size of the problem. The CPU times are in seconds. 41

4.4 Medium-scale case studies for the symmetric TSP. The experiments were repeated ten times and the average minimum and maximum of $S(\mathbf{x})$ were recorded. The parameters of the ADAM algorithm are $\varrho = 0.5$, $N = 10^2$ and the 2-opt updating is repeated $b = n \times 50$ times, where n is the size of the problem. The CPU times are in seconds. 43

4.5 Case studies for the symmetric QAP. The experiments were repeated ten times and the average, minimum, and maximum of $S(\mathbf{x})$ were recorded. The parameters of the ADAM algorithm are $\varrho = 0.5$, $N = 10^3$ and for the conditional sampling $b = n$, where n is the size of the problem. 45

4.6 The levels and splitting factors were computed using the ADAM algorithm with $N = 10^4$ and $\varrho = 0.01$. 49

4.7 Comparison between the EE sampler and the GS-sampler. The upper and lower halves correspond to the cases considered in Example 4.4.2 and 4.4.3, respectively. The values in the brackets are the corresponding estimated standard deviations. 59
4.8 12 SAT counting problems via IS estimator (4.14). In all cases $\varrho = 0.5$. 64

9.1 Results over 10 independent simulation experiments. In all cases the domain was assumed to be $\mathbb{R}$. 99

List of Figures

2.1 Typical evolution of the process $S(X)$. 8
2.2 Typical evolution of the splitting algorithm for a two-dimensional Markov process $\{(X_u^{(1)}, X_u^{(2)}), u \geqslant 0\}$. 9

3.1 Typical evolution of the GS algorithm in a two dimensional state space. 12
3.2 Typical evolution of $S(\mathbf{X})$ corresponding to the scenario in Figure 3.1. 12
3.3 Typical evolution of the FE-GS algorithm in a two dimensional state space. . . 25
3.4 Typical evolution of $S(\mathbf{X})$ corresponding to the scenario in Figure 3.3. 25

4.1 Plot of the pdf $h(\mathbf{z})/\mathscr{Z}$ for $\lambda = 12$. 33
4.2 Left: Plot of the true $\mathscr{Z}(\lambda)$ (solid) versus the estimate $\widehat{\mathscr{Z}}(\lambda)$ (dashed). Right: Plot of the estimated relative error, computed using (4.7). 49
4.3 The empirical distribution of $\mathbf{Z}$ conditional on $S(\mathbf{X}) \geqslant \gamma_t$ for $t = 0, 1, 3, 6$, respectively. 55
4.4 Empirical distribution of the output of the standard Gibbs sampler. Here the chain of length 10^9 is thinned to have 10^3 points. 56
4.5 Kernel density estimate of the final population (12800 points). 59
4.6 A histogram constructed from the empirical distribution of the total magnetization $M = \sum_{n=1}^{50} Z_n$. 61

7.1 Boundary bias in the neighborhood of $x = 0$. 76

8.1 Small and large bandwidth behavior of the diffusion density in Example 2. . . . 81

9.1 The new bandwidth selection rule in Algorithm 9.1.1 leads to improved performance compared to old plug-in rules. 91
9.2 A two-dimensional example with 600 points generated uniformly within the ellipse $\mathscr{X} = \{\mathbf{x} : x_1^2 + (4x_2)^2 = 4\}$. 100

Part I

The Generalized Splitting Method for Combinatorial Counting and Rare-Event Probability Estimation

Chapter 1

Introduction to Part One

In this chapter we motivate the need for a generalized version of the original splitting method. We explain the setting in which the generalized splitting method is applied and how it differs from the original splitting method.

One of the first Monte Carlo techniques for rare-event probability estimation is the *splitting method*, proposed by Kahn and Harris [41]. In the splitting technique, sample paths of a Markov process are split into multiple copies at various stages of the simulation, with the objective of generating more occurrences of the rare event. The rare event is represented as the intersection of a nested sequence of events, and the probability of the rare event is thus the product of conditional probabilities, each of which can be estimated much more accurately than the rare-event probability itself. Applications of the splitting method in this Markovian setting arise in particle transmission [41], queueing systems [24, 25, 23], and reliability [50]. The method has been given new impetus by the RESTART (REpetitive Simulation Trials After Reaching Thresholds) method [76, 78, 77], and has gradually evolved [27, 28] to become an effective simulation technique for dynamic simulation models. Recent improvements include the adaptive selection of the splitting levels [12] and the use of quasi Monte Carlo estimators [51].

The aim of this thesis is to introduce a new algorithm, called the *Generalized Splitting* (GS) algorithm, which extends the applicability of the splitting method to both static (that is, time-independent) and non-Markovian models. The earliest version of the GS method is described in [6]. In contrast to the specific Markov setting of the classical splitting method, the GS method involves the following general framework ([67]): let ℓ be the expected performance of a stochastic system, of the form

$$\ell = \mathbb{E}_f[H(\mathbf{X})] = \int f(\mathbf{x})H(\mathbf{x})\mu(d\mathbf{x}), \tag{1.1}$$

where H is a real-valued function, $\mathbf{X}$ is a random vector, and f the density of $\mathbf{X}$ with respect to some base measure μ. A special case of (1.1) is obtained when $H(\mathbf{x}) = I\{S(\mathbf{x}) \geqslant \gamma\}$, where

S is a score function and γ a parameter large enough such that

$$\ell = \ell(\gamma) = \mathbb{E}_f[I\{S(\mathbf{X}) \geqslant \gamma\}] = \mathbb{P}_f(S(\mathbf{X}) \geqslant \gamma) \tag{1.2}$$

is very small, so that $\ell(\gamma)$ is a rare-event probability [67, 68]. The subscript f in (1.2) indicates that the expectation and probability are taken with respect to the density f. Another special case of (1.1) is obtained when $H(\mathbf{x}) = \mathrm{e}^{-S(\mathbf{x})/\gamma}$, which arises frequently in statistical mechanics in the estimation of the so-called partition function [66].

Using the GS algorithm, we construct unbiased estimators for rare-event probabilities of the form (1.2) — and, in general, multidimensional integrals of the form (1.1). In addition, the method provides unbiased estimates for the variances of the estimators. The GS method tackles these static non-Markovian problems by artificially constructing a Markov chain using, for example, Gibbs or Metropolis-Hastings moves, and then applying the splitting idea to the Markov process induced by these moves.

The GS algorithm has the following advantages over existing MCMC algorithms for estimating (1.1). First, the GS algorithm provides an unbiased estimator $\widehat{\ell}$ for ℓ in (1.1) without the need for a burn-in period. In other words, it is not necessary that the Markov chain constructed by the algorithm reach stationarity or mix well in order to obtain unbiased and consistent estimates for ℓ. Second, unlike most MCMC algorithms, the GS algorithm provides a consistent and unbiased estimate of the mean square error of $\widehat{\ell}$. Third, there are no problems associated with selecting appropriate starting values for the Markov chain in the GS algorithm. Moreover, we will provide examples where the Markov chain constructed by the GS algorithm converges faster than standard MCMC algorithms and as well as recent algorithms such as the Equi-energy sampler [47]. Finally, while the stationarity of the chain constructed by the GS algorithm is not essential for the estimation of ℓ within the GS framework, testing the hypothesis that the chain has reached stationarity is easy and computationally inexpensive. These properties allow for substantial computational savings over traditional MCMC algorithms. Further, we show that the GS algorithm can be used to significantly improve importance sampling methods, such as the Cross Entropy (CE) method [67], for the estimation of (1.1).

The rest of the first part of the thesis is organized as follows. In chapter 2 we review the classical splitting method. In chapter 3 we explain how to obtain unbiased estimates of (1.2) using the GS methodology. We apply the method to the satisfiability (SAT) counting problem — a notoriously difficult combinatorial counting problem. We prove the unbiasedness property of the GS algorithm and explain the differences between the classical splitting method and the GS method. In chapter 4, section 4.1 we extend the applicability of the algorithm to the more general problem of estimating (1.1). In section 4.2 we use the algorithm as an optimization

routine. In section 4.3 we apply the method in the context of sensitivity analysis and estimate the partition function in the Ising model. In section 4.4 we show how the algorithm can be used for sampling from multidimensional densities for which the standard MCMC methods fail. Finally, in section 4.5 we use the GS algorithm in combination with some importance sampling to obtain highly reliable estimates for the SAT counting problem. In conclusion, chapter 5 summarizes the findings and gives possible directions for future research.

Chapter 2

Classical Splitting for Dynamic Simulation

In this chapter we provide background material about the classical splitting technique. We will refer back to this chapter when we compare the novel generalized splitting algorithm with the classical version.

A basic description of the classical splitting method is as follows. Consider a Markov process $X := \{X_u,\, u \geqslant 0\}$ with state space $\mathscr{X} \subseteq \mathbb{R}^n$, and let $S(\,\cdot\,)$ be a real-valued measurable function on $\mathscr{X}$, referred to as the *score* function. Assume for definiteness that $S(X_0) = 0$. For any *threshold* or *level* $\gamma > 0$, let U_γ denote the first time that the process $S := \{S(X_u),\, u \geqslant 0\}$ hits the set $[\gamma, \infty)$; and let U_0 denote the first time after 0 that S hits the set $(-\infty, 0]$. We assume that U_γ and U_0 are well-defined finite stopping times with respect to the history of X. One then is interested in the probability, ℓ, of the event $E_\gamma := \{U_\gamma < U_0\}$; i.e., the probability that S up-crosses level γ before it down-crosses level 0. Note that ℓ depends on the initial distribution of X. The splitting method [24, 27] is based on the observation that if $\gamma_2 > \gamma_1$, then $E_{\gamma_2} \subset E_{\gamma_1}$. Therefore, we have by the product rule of probability that $\ell = c_1\, c_2$, with $c_1 = \mathbb{P}(E_{\gamma_1})$ and $c_2 = \mathbb{P}(E_{\gamma_2} \,|\, E_{\gamma_1})$. In many cases, estimation of $c_1 c_2$ by estimating c_1 and c_2 separately is more efficient than the direct Crude Monte Carlo (CMC) estimation of ℓ. Moreover, the same arguments may be used when the interval $[0, \gamma]$ is subdivided into *multiple* subintervals $[\gamma_0, \gamma_1), [\gamma_1, \gamma_2), \ldots, [\gamma_{T-1}, \gamma]$, where $0 = \gamma_0 < \gamma_1 < \cdots < \gamma_T = \gamma$. Again, let E_{γ_t} denote the event that the process S reaches level γ_t before down-crossing level 0. Since $E_{\gamma_0}, E_{\gamma_1}, \ldots, E_{\gamma_T}$ is a nested sequence of events, denoting $c_t = \mathbb{P}(E_{\gamma_t} \,|\, E_{\gamma_{t-1}})$, we obtain $\ell = \prod_{t=1}^{T} c_t$.

The estimation of each c_t is performed in the following way. At stage $t = 1$ we run $N_0 \times s_1$ (a fixed number) of independent copies of X and evolve the corresponding process $S(X)$. Each copy of X is run until $S(X)$ either hits the set $(-\infty, 0]$ or up-crosses the level γ_1; that is, each copy is run for a time period equal to $\min\{U_{\gamma_1}, U_0\}$. The number s_1 is an integer referred to as the *splitting factor* at stage $t = 1$. Define I_j^1 to be the indicator that the j-th copy of $S(X)$ hits the set $[\gamma_1, \infty)$ before $(-\infty, 0]$, $j = 1, \ldots, N_0 \times s_1$, and let N_1 be the total number of copies that up-cross γ_1; that is,

$$N_1 = \sum_{j=1}^{N_0 \times s_1} I_j^1.$$

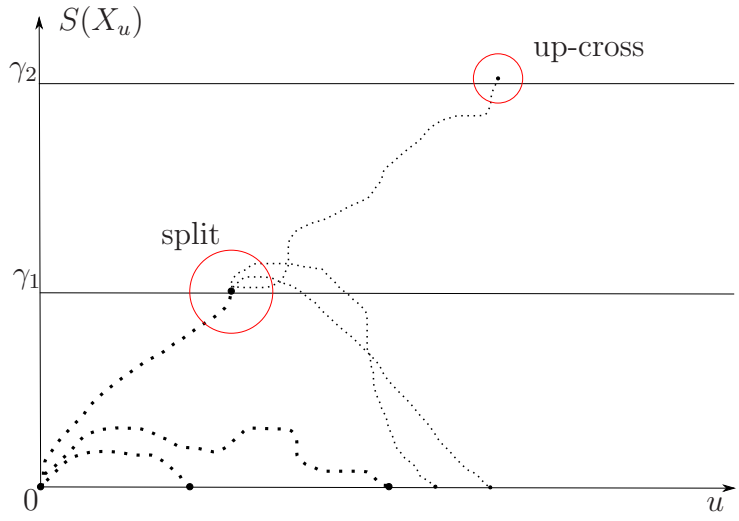

Figure 2.1: Typical evolution of the process $S(X)$.

An unbiased estimate for c_1 is $\widehat{c}_1 := \frac{N_1}{N_0 \times s_1}$. For every realization of $S(X)$ which up-crosses γ_1, we store in memory the corresponding state X_τ at the time τ of crossing. Such a state is referred to as the *entrance state* [23]. In the next stage, $t = 2$, we start s_2 new independent copies of the chain X from each of the N_1 entrance states, giving a total of $N_1 \times s_2$ new chains. Again, if we let I_j^2 indicate whether the j-th copy of X (starting from an entrance state at level γ_1) hits the set $[\gamma_2, \infty)$ before $(-\infty, 0]$, then $\widehat{c}_2 := \frac{N_2}{N_1 \times s_2}$, where $N_2 = \sum_{j=1}^{N_1 \times s_2} I_j^2$, is an unbiased estimate of c_2 [23]. This process is repeated for each subsequent $t = 3, \ldots, T$, such that $N_{t-1} \times s_t$ is the *simulation effort* at stage t, and N_t is the number of entrance states at stage t. The indicators $\{I_j^t\}$ at stage t are usually dependent, and hence the success probabilities $\{\mathbb{P}(I_j^t = 1)\}$ depend on the entrance state from which a copy of the chain X is started. It is well known [23, 27] that despite this dependence, the estimator

$$\widehat{\ell} = \prod_{t=1}^{T} \widehat{c}_t = \frac{N_T}{N_0} \prod_{t=1}^{T} s_t^{-1}$$

is unbiased. The idea of the splitting method is illustrated in Figure 2.1, where three level sets $\{\mathbf{x} : S(\mathbf{x}) = \gamma_t\}$, $t = 0, 1, 2$ are plotted. Here three independent paths of the process $S(X)$ are started from level $\gamma_0 = 0$, two of these paths *die out* by down-crossing level 0, one of the paths up-crosses level γ_1. Three new independent copies of the chain are started from the entrance state at level γ_1 (encircled on the graph), two of these copies down-cross 0, but one copy up-crosses level γ_2. Figure 2.2 shows a typical realization of a two-dimensional Markov process $\{(X_u^{(1)}, X_u^{(2)}), u \geqslant 0\}$ that corresponds to the scenario described on Figure 2.1.

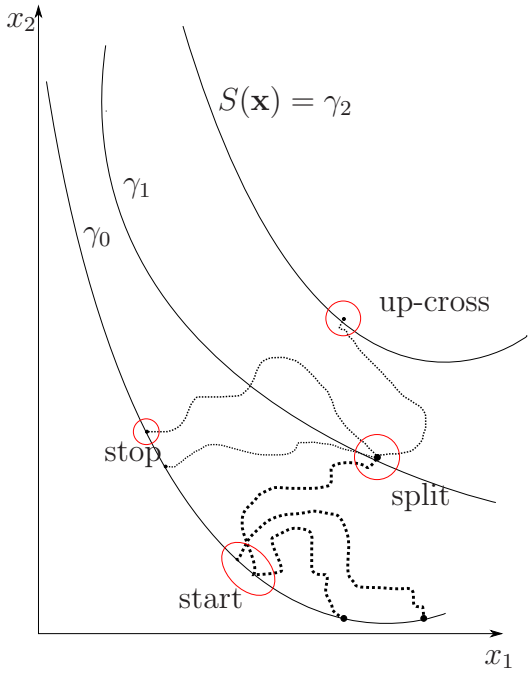

Figure 2.2: Typical evolution of the splitting algorithm for a two-dimensional Markov process $\{(X_u^{(1)}, X_u^{(2)}), u \geqslant 0\}$.

The efficiency of the splitting method depends crucially on the number of levels T, the choice of the intermediate levels $\gamma_1, \ldots, \gamma_{T-1}$, and the splitting factors $s_1, \ldots, s_T$. Ideally one would select the levels so that the conditional probabilities $\{c_t\}$ are not too small and easily estimated via CMC. Note that the total simulation effort is a random variable with expected value

$$\sum_{t=1}^{T} s_t \mathbb{E}[N_{t-1}] = N_0 \sum_{t=1}^{T} s_t \, \ell(\gamma_{t-1}) \prod_{j=1}^{t-1} s_j = N_0 \sum_{t=1}^{T} s_t \prod_{j=1}^{t-1} c_j \, s_j = N_0 \sum_{t=1}^{T} \frac{1}{c_t} \prod_{j=1}^{t} c_j \, s_j. \tag{2.1}$$

An inappropriate choice of the splitting factors may lead to an exponential growth of the simulation effort. For example, if $c_j s_j = a > 1, \forall j$, then $N_0 \sum_{t=1}^{T} \frac{1}{c_t} a^t$ grows exponentially in T. This is referred to as an *explosion* in the splitting literature [28]. Alternatively, if $c_j s_j = a < 1, \forall j$, then $\mathbb{E}[N_T] = N_0 \, a^T$ decays exponentially and with high probability N_T, and hence $\widehat{\ell}$, will be 0, making the algorithm inefficient. It is thus desirable that $c_j s_j = 1, \forall j$, that is, the splitting is at *critical value* [28]. In practice, one obtains rough estimates $\{\varrho_j\}$ of $\{c_j\}$ via a pilot run and then initializes from each entrance state $j = 1, \ldots, N_t$, at every stage t, $s_t = \varrho_t^{-1}$ paths. In case $1/\varrho_j$ is not an integer, one can generate a Bernoulli random variable with success

probability $\varrho_j^{-1} - \lfloor \varrho_j^{-1} \rfloor$ and then add it to $\lfloor \varrho_j^{-1} \rfloor$ to obtain a random integer-valued splitting factor $\mathcal{S}_j$ with expected value $1/\varrho_j$ [28]. This version of the splitting algorithm is called the *Fixed Splitting* (FS) implementation, because at every stage t one generates a fixed expected number of copies ϱ_t^{-1} from each entrance state. An alternative to the FS implementation is the *Fixed Effort* (FE) implementation, where the simulation effort is fixed to N at each stage, instead of the number of copies [23]. The estimator then is

$$\widehat{\ell}_{\mathrm{FE}} = \prod_{t=1}^{T} \frac{N_t}{N}.$$

The FE implementation prevents explosions in the number of total Markov chain copies, but has the disadvantage that it is more difficult to analyze the variance of $\widehat{\ell}_{\mathrm{FE}}$ [23, 24]. Having given a brief overview of the classical splitting method, in the next chapter we proceed to describe the novel generalized version of the splitting method.

Chapter 3

Generalized Splitting Algorithms

In this chapter we present the Generalized Splitting method for rare-event probability estimation and combinatorial counting.

3.1 Generalized Splitting Method

We now explain how one can obtain unbiased estimates of the rare-event probability (1.2) using the splitting idea described in the previous chapter. First, partition the interval $(-\infty, \gamma]$ using intermediate levels $-\infty = \gamma_0 \leqslant \gamma_1 \leqslant \ldots \leqslant \gamma_{T-1} \leqslant \gamma_T = \gamma$. Note that, unlike in the classical splitting, $\gamma_0 = -\infty$. We assume that the sequence of levels is chosen such that the conditional probabilities $\mathbb{P}_f(S(\mathbf{X}) \geqslant \gamma_t \,|\, S(\mathbf{X}) \geqslant \gamma_{t-1}) = c_t,\ t = 1, \ldots, T$, are not rare-event probabilities, and that we have rough estimates $\{\varrho_t\}$ of $\{c_t\}$ available. We will later explain how we can construct the sequence $\{\gamma_t, \varrho_t\}_{t=1}^T$ using the adaptive pilot algorithm described in [6]. Without loss of generality, we assume that generating random variables from f is easy.

Before giving the full details in Algorithm 3.1.1, we illustrate the GS recipe on a typical problem of the form (1.2) with three levels ($T = 3$). Figure 3.1 depicts the GS algorithm as applied to a particular two-dimensional rare-event simulation problem. The three level sets $\{\mathbf{x} : S(\mathbf{x}) = \gamma_t\},\ t = 1, 2, 3$ are plotted as convex curves and the entrance states at each stage are encircled. We assume that the $\{\gamma_t\}$ are given and that $\varrho_t = 1/10$ for all t; that is, the splitting factors are $s_t = \varrho_t^{-1} = 10$ for all t. Initially, at stage $t = 1$, we generate $N_0/\varrho_1 = 10$ independent points from the density $f(\mathbf{x})$. We denote the points $\mathbf{X}_1, \ldots, \mathbf{X}_{10}$. Two of these points, namely $\mathbf{X}_1$ and $\mathbf{X}_2$, are such that both $S(\mathbf{X}_1)$ and $S(\mathbf{X}_2)$ are above the γ_1 threshold. Points $\mathbf{X}_1$ and $\mathbf{X}_2$ are thus the entrance states for the next stage of the algorithm, and $N_1 = \sum_{j=1}^{10} I\{S(\mathbf{X}_j) \geqslant \gamma_1\} = 2$. In stage $t = 2$ we start independent Markov chains from each of the entrance states $\mathbf{X}_1$ and $\mathbf{X}_2$. The only requirement is that each Markov chain has a stationary distribution equal to the conditional distribution of $\mathbf{X}$ given that $S(\mathbf{X}) \geqslant \gamma_1$, where $\mathbf{X} \sim f$. The length of both chains is equal to $s_2 = 10$.

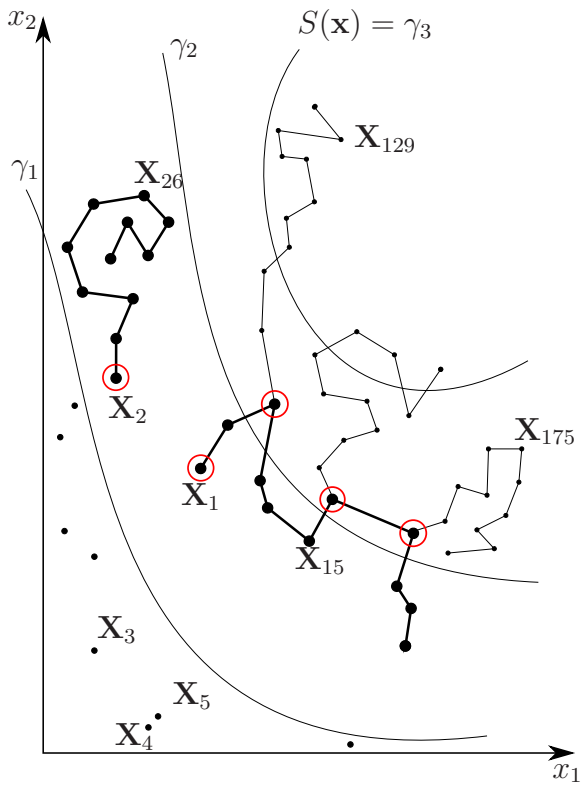

Figure 3.1: Typical evolution of the GS algorithm in a two dimensional state space.

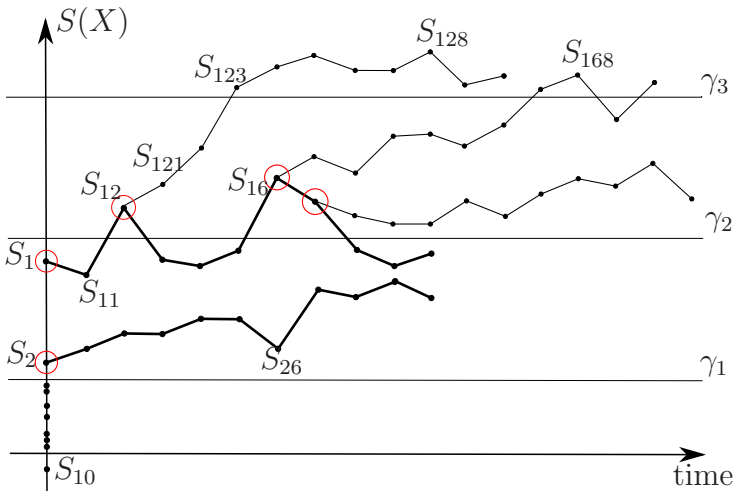

Figure 3.2: Typical evolution of $S(\mathbf{X})$ corresponding to the scenario in Figure 3.1.

Thus, the simulation effort for $t = 2$ is $N_1 \times s_2 = 20$. In other words, in stage $t = 2$ we generate

$$\mathbf{X}_{ij} \sim \kappa_1(\mathbf{x} \,|\, \mathbf{X}_{i,j-1}), \quad j = 1, \ldots, 10, \;\; i = 1, 2,$$

where $\mathbf{X}_{i,0} = \mathbf{X}_i$, and $\kappa_1(\,\cdot \mid \cdot\,)$ is a Markov transition density with stationary pdf f_1 given by $(t = 1)$

$$f_t(\mathbf{x}) = \frac{f(\mathbf{x}) I\{S(\mathbf{x}) \geqslant \gamma_t\}}{\ell(\gamma_t)}. \tag{3.1}$$

Figure 3.1 depicts the Markov chains as branches sprouting from points $\mathbf{X}_1$ and $\mathbf{X}_2$. Note that these branches are drawn thicker than branches generated at state $t = 3$. None of the points on the $\mathbf{X}_2$ branch have a score above γ_2. Points $\mathbf{X}_{12}, \mathbf{X}_{16}, \mathbf{X}_{17}$ from the $\mathbf{X}_1$ branch (encircled) make it above the γ_2 threshold. These three points will be the entrance states for stage $t = 3$ of the algorithm. Thus,

$$N_2 = \sum_{j=1}^{10} \Big(I\{S(\mathbf{X}_{1j}) \geqslant \gamma_2\} + I\{S(\mathbf{X}_{2j}) \geqslant \gamma_2\} \Big) = 3.$$

In the final stage we start three independent Markov chains with stationary density f_2 from each of the entrance states $\mathbf{X}_{12}, \mathbf{X}_{16}, \mathbf{X}_{17}$. The length of each chain is $s_3 = 10$. Thus, the simulation effort for stage $t = 3$ is $3 \times 10 = 30$, and we generate

$$\mathbf{X}_{1jk} \sim \kappa_2(\mathbf{x} \,|\, \mathbf{X}_{1,j,k-1}), \quad k = 1, \ldots, 10, \;\; j = 2, 6, 7,$$

where $\mathbf{X}_{1j\,0} = \mathbf{X}_{1j}$, and $\kappa_2(\,\cdot \mid \cdot\,)$ is Markov transition density with stationary density f_2 defined in (3.1). Figure 3.1 shows that the points that up-cross level γ_3 are $\mathbf{X}_{12k}$, $k = 3, \ldots, 10$ and $\mathbf{X}_{16k}$, $k = 7, 8, 10$. Thus, in the last stage $T = 3$ we have $N_T = 11$. Finally, an estimator of $\ell(\gamma_3)$ is

$$\widehat{\ell}(\gamma_T) = \frac{N_T}{N_0} \prod_{t=1}^{T} s_t^{-1},$$

and this gives the estimate 11×10^{-3}. We will prove later in this section that such an estimator is unbiased. Figure 3.2 shows the behavior of the score process $S(\mathbf{X})$ for every chain starting from the entrance states $\mathbf{X}_1, \mathbf{X}_2$ and $\mathbf{X}_{12}, \mathbf{X}_{16}, \mathbf{X}_{17}$ (encircled). Here $S_{ijk} = S(\mathbf{X}_{ijk})$ and time is measured in terms of the number of Markov chain moves. Note that the Markov chain paths generated at stage $t = 2$ are drawn thicker. The three chains starting from $\mathbf{X}_{12}, \mathbf{X}_{16}, \mathbf{X}_{17}$ are all dependent, because they share a common history, namely, the branch with root at $\mathbf{X}_1$. These three chains, however, are conditionally independent given the branch with root at $\mathbf{X}_1$.

Remark 3.1.1 (Non-integer splitting factors) If ϱ_t^{-1} is not an integer, then, similar to the classical splitting method, the length of each Markov chain started from an entrance state at stage t is taken to be a random integer-valued splitting factor with expected value ϱ_t^{-1}. Note that even though all entrance states at stage t are assigned a random splitting factor, these factors are independent and have the same expected value. Such random splitting factors are conveniently constructed by generating independent Bernoulli random variables with a common success probability $\varrho_t^{-1} - \lfloor \varrho_t^{-1} \rfloor$ and then adding $\lfloor \varrho_t^{-1} \rfloor$ to the Bernoulli variables.

We now describe the GS method as applied to the problem of estimating (1.2).

Algorithm 3.1.1 (GS algorithm for estimating $\ell = \mathbb{P}_f(S(\mathbf{X}) \geqslant \gamma)$)
Given a sequence $\{\gamma_t, \varrho_t\}_{t=1}^T$ and a sample size N, execute the following steps.

1. **Initialization.** *Set $t = 1$ and $N_0 = \varrho_1 \left\lfloor \frac{N}{\varrho_1} \right\rfloor$ (which ensures that N_0/ϱ_1 is an integer). Generate*

$$\mathbf{X}_1, \ldots, \mathbf{X}_{N_0/\varrho_1} \sim_{iid} f(\mathbf{x})$$

and denote $\mathcal{X}_0 = \{\mathbf{X}_1, \ldots, \mathbf{X}_{N_0/\varrho_1}\}$. Let $\mathcal{X}_1 = \{\mathbf{X}_1, \ldots, \mathbf{X}_{N_1}\}$ be the largest subset of elements in $\mathcal{X}_0$ for which $S(\mathbf{X}) \geqslant \gamma_1$ (points $\mathbf{X}_1$ and $\mathbf{X}_2$ on Figure 3.1). Here, N_1 is the random number of vectors in $\mathcal{X}_0$ such that $S(\mathbf{X}) \geqslant \gamma_1$.

2. **Markov chain sampling.** *For each $\mathbf{X}_i$ in $\mathcal{X}_t = \{\mathbf{X}_1, \ldots, \mathbf{X}_{N_t}\}$, sample independently:*

$$\mathbf{Y}_{ij} \sim \kappa_t(\mathbf{y} \,|\, \mathbf{Y}_{i,j-1}), \quad \mathbf{Y}_{i,0} = \mathbf{X}_i, \quad j = 1, \ldots, \mathcal{S}_{ti}, \tag{3.2}$$

where

$$\mathcal{S}_{ti} - \left\lfloor \frac{1}{\varrho_{t+1}} \right\rfloor \sim_{iid} \mathsf{Ber}\left(\frac{1}{\varrho_{t+1}} - \left\lfloor \frac{1}{\varrho_{t+1}} \right\rfloor \right), \quad i = 1, \ldots, N_t.$$

Here $\kappa_t(\mathbf{y} \,|\, \mathbf{Y}_{i,j-1})$ is a Markov transition density with stationary pdf

$$f_t(\mathbf{y}) = \frac{f(\mathbf{y}) I\{S(\mathbf{y}) \geqslant \gamma_t\}}{\ell(\gamma_t)}.$$

Each $\mathcal{S}_{ti}$ is a splitting factor equal to $\left\lfloor \frac{1}{\varrho_{t+1}} \right\rfloor$ plus a Bernoulli random variable with success probability $\frac{1}{\varrho_{t+1}} - \left\lfloor \frac{1}{\varrho_{t+1}} \right\rfloor$. Reset

$$\mathcal{X}_t := \Big\{ \mathbf{Y}_{11}, \mathbf{Y}_{12}, \ldots, \mathbf{Y}_{1,\mathcal{S}_{t1}}, \cdots\cdots\cdots, \mathbf{Y}_{N_t 1}, \mathbf{Y}_{N_t 2}, \ldots, \mathbf{Y}_{N_t, \mathcal{S}_{t N_t}} \Big\},$$

where $\mathcal{X}_t$ contains $|\mathcal{X}_t| = \sum_{i=1}^{N_t} \mathcal{S}_{t\,i}$ elements and $\mathbb{E}[\,|\mathcal{X}_t| \mid N_t\,] = \frac{N_t}{\varrho_{t+1}}$.
For example, on Figure 3.1 we have $\mathcal{X}_1 = \{\mathbf{X}_{ij},\ i = 1, 2; j = 1, \ldots, 10\}$ and $|\mathcal{X}_1| = 20$.

3. Updating. *Let $\mathcal{X}_{t+1} = \{\mathbf{X}_1, \ldots, \mathbf{X}_{N_{t+1}}\}$ be the largest subset of elements in $\mathcal{X}_t$ for which $S(\mathbf{X}) \geqslant \gamma_{t+1}$. Here, N_{t+1} is the random number of vectors in $\mathcal{X}_t$ such that $S(\mathbf{X}) \geqslant \gamma_{t+1}$ (e.g., $\mathcal{X}_2 = \{\mathbf{X}_{1j}, j = 2, 6, 7\}$ and $N_2 = 3$ on Figure 3.1). Reset the counter $t := t + 1$.*

4. Stopping condition. *If $t = T$ or $N_t = 0$, set $N_{t+1} = N_{t+2} = \cdots = N_T = 0$ and go to Step 5; otherwise, repeat from Step 2.*

5. Final estimator. *Deliver the unbiased estimate of the rare-event probability:*

$$\widehat{\ell} = \frac{N_T}{N_0} \prod_{t=1}^{T} \varrho_t, \tag{3.3}$$

and the unbiased estimate of the variance:

$$\widehat{\mathrm{Var}(\widehat{\ell})} = \frac{\prod_{t=1}^{T} \varrho_t^2}{N_0(N_0 - \varrho_1)} \sum_{i=1}^{N_0/\varrho_1} \left(O_i - \frac{\varrho_1}{N_0} N_T \right)^2, \tag{3.4}$$

where O_i denotes the number of points that share a common history with the $i-$th point in the initial population $\mathcal{X}_0$ and are above $\gamma_T = \gamma$ on the final $t = T$ stage. For example, on Figure 3.1 we have $O_1 = 11$ and $O_i = 0$ for $i = 2, \ldots, 10$, since only points $\mathbf{X}_{12k}$, $k = 3, \ldots, 10$ and $\mathbf{X}_{16k}$, $k = 7, 8, 10$ are above γ_3 threshold and they are all part of a branch that has $\mathbf{X}_1$ at its root.

In Step 2 of Algorithm 3.1.1 a move from $\mathbf{X}$ to $\mathbf{Y}$ using the transition density $\kappa_t(\mathbf{y} \,|\, \mathbf{x})$ can, for example, consist of drawing $\mathbf{Y}$ from the conditional pdf

$$Y_i \sim f_t(y_i \,|\, Y_1, \ldots, Y_{i-1}, X_{i+1}, \ldots, X_n), \quad i = 1, \ldots, n,$$

as in the Gibbs sampling method [68]. The transition density is then

$$\kappa_t(\mathbf{y} \,|\, \mathbf{x}) = \prod_{i=1}^{n} f_t(y_i \,|\, y_1, \ldots y_{i-1}, x_{i+1}, \ldots, x_n). \tag{3.5}$$

Alternatively, a move from $\mathbf{X}$ to $\mathbf{Y}$ may consist of a Metropolis-Hastings (MH) move (see, e.g., [68]):

$$\mathbf{Y} = \begin{cases} \mathbf{Y}^*, & \text{if} \quad U \leqslant \rho(\mathbf{X}, \mathbf{Y}^*) \\ \mathbf{X}, & \text{otherwise} \end{cases},$$

where

$$\rho(\mathbf{x}, \mathbf{y}) = \min \left\{ \frac{f(\mathbf{y}) I\{S(\mathbf{y}) \geqslant \gamma_t\} q(\mathbf{x} \,|\, \mathbf{y})}{f(\mathbf{x}) I\{S(\mathbf{x}) \geqslant \gamma_t\} q(\mathbf{y} \,|\, \mathbf{x})}, \; 1 \right\},$$

$q(\cdot \mid \cdot)$ is a MH proposal density, and $\mathbf{Y}^* \sim q(\mathbf{y}^* \mid \mathbf{X})$ and $U \sim \mathsf{U}(0,1)$ independently. In other words, the transition density is:

$$\kappa_t(\mathbf{y} \mid \mathbf{x}) = \rho(\mathbf{x}, \mathbf{y}) q(\mathbf{y} \mid \mathbf{x}) + \delta(\mathbf{x} - \mathbf{y}) \left(1 - \int \rho(\mathbf{x}, \mathbf{y}) q(\mathbf{y} \mid \mathbf{x}) d\mathbf{y}\right),$$

where $\delta(\cdot)$ denotes the Dirac delta function.

Although Algorithm 3.1.1 has strong similarities with the classical Fixed Splitting method described in the introduction, there are some important differences. First, as seen from Figure 3.1, during the initialization of the GS algorithm we do not run Markov chains, but generate iid vectors from the density f in (1.2). In contrast, in the classical Fixed Splitting algorithm one always initializes with a population of independent Markov chains. Second, in Algorithm 3.1.1 the first level is always $\gamma_0 = -\infty$, while in the classical splitting algorithm γ_0 is usually finite. More importantly, while in the classical Fixed Splitting we run the *same* Markov process $X = \{X_u, u \geqslant 0\}$ throughout all stages of the algorithm, in the GS algorithm the stationary distribution of the Markov chains *changes* across the stages. More precisely, in the GS algorithm, the stationary distribution at stage t is f_{t-1}. As a result of this, *no Markov chain paths can go below level* γ_{t-1} *in stage* t of the GS algorithm. In contrast, the paths in the classical Fixed Splitting can down-cross thresholds and down-cross level γ_0. This difference is illustrated in Figures 2.1 and 3.2. While there are two paths in Figure 2.1 that down-cross γ_0 in stage $t = 2$, there are no paths in Figure 3.2 that down-cross levels γ_1 and γ_2 in stages $t = 2$ and $t = 3$ respectively.

As an application of the algorithm we consider the following counting problem.

Example 3.1.1 (SAT Counting Problem) The Boolean Satisfiability problem (SAT) is a central problem in combinatorial optimization. Any NP complete problem, such as the Traveling Salesman Problem, can be redefined in polynomial time into a SAT problem. The SAT problem also arises in the context of computer architecture design, image processing and scheduling. There are many different mathematical formulations of the SAT problem [29]. Here we use a formulation which is convenient to use on the problems from the SATLIB website `www.satlib.org`. Let $\mathbf{x} = (x_1, \ldots, x_n)'$, $x_i \in \{0, 1\}$ denote a so-called *truth assignment*. Let $A = (A_{ij})$ denote a $m \times n$ *clause matrix*, that is, all elements of A belong to the set $\{-1, 0, 1\}$, and define $\mathbf{b} = [b_1, \ldots, b_m]'$ to be the vector with entries $b_i = 1 - \sum_{j=1}^{n} I\{A_{ij} = -1\}$. In the standard SAT problem one is interested in finding a truth assignment $\mathbf{x}$ for which $A\mathbf{x} \geqslant \mathbf{b}$. In the SAT *counting* problem, one is interested in finding the *total number* of truth assignments $\mathbf{x}$ for which $A\mathbf{x} \geqslant \mathbf{b}$. The SAT counting problem is considered more complex than the SAT problem [68, 80], and in fact the SAT counting problem is known to be a notoriously difficult #P complete problem. Here we aim to find the total number of solutions of the SAT problem,

that is, we wish to estimate the size of the set

$$\mathscr{X}^* = \left\{ \mathbf{x} : \sum_{i=1}^{m} I_{\left\{\sum_{j=1}^{n} A_{ij} x_j \geqslant b_i\right\}} \geqslant m \right\}.$$

To estimate $|\mathscr{X}^*|$ we consider the problem of estimating the probability

$$\ell = \mathbb{P}\left(S(\mathbf{X}) \geqslant m\right), \quad \{X_j\} \sim_{\text{iid}} \mathsf{Ber}(1/2), \quad S(\mathbf{x}) = \sum_{i=1}^{m} I\left\{ \sum_{j=1}^{n} A_{ij} x_j \geqslant b_i \right\}, \tag{3.6}$$

via Algorithm 3.1.1. The size of the set is then estimated from the relation $|\mathscr{X}^*| = \ell \times 2^n$. As a numerical example, consider the `uf20-91` ($n = 20$, $m = 91$) problem with instance `uf20-01` from the SATLIB website. We applied Algorithm 3.1.1 using the levels and weights given in Table 3.1 and sample size $N = 10^4$. The Markov transition density κ_t in Step 2 of Algorithm 3.1.1 is given by (3.5) and the stationary pdf is

$$f_t(\mathbf{x}) = \frac{1}{2^n \ell(\gamma_t)} I\left\{ \sum_{i=1}^{m} I\left\{ \sum_{j=1}^{n} A_{ij} x_j \geqslant b_i \right\} \geqslant \gamma_t \right\}, \quad x_j \in \{0, 1\}.$$

In other words, the action of the transition density $\kappa_t(\mathbf{y} \,|\, \mathbf{x})$ is equivalent to the following Gibbs sampling procedure.

1. Given a state $\mathbf{x}$ such that $S(\mathbf{x}) \geqslant \gamma_t$, generate $Y_1 \sim f_t(y_1 \,|\, x_2, \ldots, x_n)$;
2. For each $k = 2, \ldots, n-1$, generate $Y_k \sim f_t(y_k \,|\, Y_1, \ldots, Y_{k-1}, x_{k+1}, \ldots, x_n)$;
3. Finally, generate $Y_n \sim f_t(y_n \,|\, Y_1, \ldots, Y_{n-1})$.

Note that one can write the conditional density of Y_k as

$$f_t(y_k \,|\, Y_1, \ldots, Y_{k-1}, x_{k+1}, \ldots, x_n) = \begin{cases} p_k, & y_k = 1 \\ 1 - p_k, & y_k = 0 \end{cases},$$

$$\text{where} \qquad p_k = \frac{I\{S_{k_1} \geqslant \gamma_t\}}{I\{S_{k_1} \geqslant \gamma_t\} + I\{S_{k_0} \geqslant \gamma_t\}},$$

$$S_{k_0} = \sum_{i=1}^{m} I\left\{ \sum_{j=1}^{k-1} A_{ij} Y_j + \sum_{j=k+1}^{n} A_{ij} x_j \geqslant b_i \right\},$$

$$S_{k_1} = \sum_{i=1}^{m} I\left\{ \sum_{j=1}^{k-1} A_{ij} Y_j + A_{ik} + \sum_{j=k+1}^{n} A_{ij} x_j \geqslant b_i \right\}.$$

With the above setup we obtained an estimate $\widehat{|\mathscr{X}^*|} = \widehat{\ell} \times 2^n = 7.90$ with an estimated

Table 3.1: The sequence of levels and splitting factors used in Algorithm 3.1.1 for instance `uf20-01`.

t	γ_t	ϱ_t
1	81	0.4635
2	84	0.3207
3	86	0.2656
4	87	0.4059
5	88	0.2747
6	89	0.1604
7	90	0.1252
8	91	0.0793

relative error of 2.7%. Direct enumeration of all possible assignments gives $|\mathscr{X}^*| = 8$. All of these solutions were part of the final sample of the GS algorithm. Similar to the classical Fixed Splitting algorithm, the total simulation effort in the GS algorithm is given by (2.1). Assuming that the splitting is at critical value, the expected simulation effort in this example is $N_0 \sum_{t=1}^{T} \frac{1}{\varrho_t} \approx 425,000$ and the actual simulation effort during the simulation is $360,000$. Total enumeration of all possible truth assignments for which $A\mathbf{x} \geqslant \mathbf{b}$ would require the equivalent of a simulation effort of size $2^{20} \approx 10^6$.

Remark 3.1.2 (Alternative formulation) There are different ways in which the SAT counting problem can be recast into a rare-event probability estimation problem. Instead of considering the rare-event probability (3.6), one can consider estimating

$$\ell = \mathbb{P}\left(\sum_{i=1}^{n} \min\left\{ \sum_{j=1}^{m} A_{ij} X_j - b_i,\ 0 \right\} \geqslant 0 \right), \quad X_j \sim_{\text{iid}} \mathsf{Ber}(1/2),$$

thus redefining $S(\mathbf{x}) = \sum_{i=1}^{n} \min\left\{ \sum_{j=1}^{m} A_{ij} x_j - b_i,\ 0 \right\}$, and $\gamma_T = \gamma = 0$. Observe that if $S(\mathbf{x}) = 0$, then $\mathbf{x}$ satisfies the constraint $A\mathbf{x} \geqslant \mathbf{b}$.

Concerning the properties of the estimator (3.3) and its variance, we have the following result.

Proposition 3.1.1 (Unbiasedness of the GS estimator) *The estimator in* (3.3) *is an unbiased estimator of* ℓ*, and* (3.4) *is an unbiased estimator of* $\mathrm{Var}(\widehat{\ell})$.

Proof: We will prove the result for $T = 3$, as the result for general T follows by a similar

argument. Using the notation of Figure 3.1, we can write

$$N_3 = \sum_{i=1}^{N_0/\varrho_1} I\{S(\mathbf{X}_i) \geqslant \gamma_1\} \sum_{j=1}^{\mathcal{S}_{1i}} I\{S(\mathbf{X}_{ij}) \geqslant \gamma_2\} \sum_{k=1}^{\mathcal{S}_{2j}} I\{S(\mathbf{X}_{ijk}) \geqslant \gamma_3\},$$

$$\begin{aligned}
\text{where} \quad & \mathbf{X}_i \sim f(\,\cdot\,), \quad i = 1, \ldots, \frac{N_0}{\varrho_1}, \text{ independently,} \\
& \mathbf{X}_{ij} \sim \kappa_1(\,\cdot \,|\, \mathbf{X}_{i,j-1}), \quad \mathbf{X}_{i0} = \mathbf{X}_i, \quad i = 1, \ldots, \frac{N_0}{\varrho_1}, \;\; j = 1, \ldots, \mathcal{S}_{1i}, \\
& \mathbf{X}_{ijk} \sim \kappa_2(\,\cdot \,|\, \mathbf{X}_{i,j,k-1}), \quad \mathbf{X}_{ij0} = \mathbf{X}_{ij}, \quad \forall i, j, k.
\end{aligned}$$

Since the splitting factors $\{\mathcal{S}_{1i}, \mathcal{S}_{2j}\}$ are independent of $\{\mathbf{X}_i, \mathbf{X}_{ij}, \mathbf{X}_{ijk}, \forall i, j, k\}$, we can write

$$\mathbb{E}[N_3 \,|\, \{\mathcal{S}_{1i}, \mathcal{S}_{2j}\}] = \sum_{i=1}^{N_0/\varrho_1} \sum_{j=1}^{\mathcal{S}_{1i}} \sum_{k=1}^{\mathcal{S}_{2j}} \mathbb{E}\left[I_{\{S(\mathbf{X}_i) \geqslant \gamma_1\}} \; I_{\{S(\mathbf{X}_{ij}) \geqslant \gamma_2\}} \; I_{\{S(\mathbf{X}_{ijk}) \geqslant \gamma_3\}} \right].$$

The expectation under the triple summation is

$$\int \cdots \int f(\mathbf{x}_i) I_{\{S(\mathbf{x}_i) \geqslant \gamma_1\}} \prod_{m=1}^{j} \kappa_1(\mathbf{x}_{im} \,|\, \mathbf{x}_{i,m-1}) I_{\{S(\mathbf{x}_{ij}) \geqslant \gamma_2\}} \prod_{l=1}^{k} \kappa_2(\mathbf{x}_{ijl} \,|\, \mathbf{x}_{ij,l-1}) I_{\{S(\mathbf{x}_{ijk}) \geqslant \gamma_3\}} \, d\mathbf{x}_i \cdots d\mathbf{x}_{ijk},$$

which by integrating in the order

$$\mathbf{x}_i, \mathbf{x}_{i1}, \ldots, \mathbf{x}_{ij}, \mathbf{x}_{ij1}, \ldots, \mathbf{x}_{ijk},$$

and each time applying the invariance property

$$\int f(\mathbf{x}) I\{S(\mathbf{x}) \geqslant \gamma_t\} \kappa_t(\mathbf{y} \,|\, \mathbf{x}) \, d\mathbf{x} = f(\mathbf{y}) I\{S(\mathbf{y}) \geqslant \gamma_t\}, \quad \forall t, \tag{3.7}$$

yields $\ell(\gamma_3) = \int f(\mathbf{x}) I\{S(\mathbf{x}) \geqslant \gamma_3\} d\mathbf{x}$. Therefore

$$\mathbb{E}[N_3] = \mathbb{E}[\mathbb{E}[N_3 \,|\, \{\mathcal{S}_{1i}, \mathcal{S}_{2j}\}]] = \mathbb{E}\left[\sum_{i=1}^{N_0/\varrho_1} \sum_{j=1}^{\mathcal{S}_{1i}} \sum_{k=1}^{\mathcal{S}_{2j}} \ell \right] = \ell \frac{N_0}{\varrho_1 \varrho_2 \varrho_3},$$

and the estimator (3.3) is unbiased.

To derive the variance of (3.3), observe that by definition

$$O_i = I\{S(\mathbf{X}_i) \geqslant \gamma_1\} \sum_{j=1}^{\mathcal{S}_{1i}} I\{S(\mathbf{X}_{ij}) \geqslant \gamma_2\} \sum_{k=1}^{\mathcal{S}_{2j}} I\{S(\mathbf{X}_{ijk}) \geqslant \gamma_3\}.$$

Table 3.2: The sequence of levels and splitting factors used in Algorithm 3.1.1 for instance `uf75-01`.

t	γ_t	ϱ_t
1	285	0.4750
2	289	0.4996
3	292	0.4429
4	294	0.4912
5	296	0.4369
6	298	0.3829
7	300	0.3277
8	302	0.2676
9	303	0.4982
10	304	0.4505
11	305	0.4359
12	306	0.4239
13	307	0.3990
14	308	0.3669
15	309	0.3658
16	310	0.3166
17	311	0.3072
18	312	0.3104
19	313	0.2722
20	314	0.2253
21	315	0.2235
22	316	0.2071
23	317	0.1892
24	318	0.1722
25	319	0.1363
26	320	0.1413
27	321	0.1237
28	322	0.0953
29	323	0.0742
30	324	0.0415
31	325	0.0115

Since the $\{\mathbf{X}_i\}$ are independent and $N_T = \sum_i O_i$, we have $\mathrm{Var}(N_T) = \frac{N_0}{\varrho_1}\mathrm{Var}(O_i)$, from which (3.4) follows. □

Example 3.1.2 As another example, consider the instance `uf75-01` ($m = 325$, $n = 75$). We applied Algorithm 3.1.1 with $N = 10^4$ and the splitting factors and levels given in Table 3.2, thus giving a total simulation effort of about 2.8×10^6 samples, including the pilot run. We use the same transition density κ_t described in Example 3.1.1. We obtained $\widehat{|\mathscr{X}^*|} = 2.31 \times 10^3$ with estimated relative error of 5.8%. Total enumeration of all possible truth assignments for which $A\mathbf{x} \geqslant \mathbf{b}$ would require the equivalent of a simulation effort of size $2^{75} \approx 3.7 \times 10^{22}$ and it hence impracticable. To achieve the same relative error using CMC would require a sample size of approximately $N = 4.8 \times 10^{21}$. Thus, we see that with a minimal amount of additional work the GS algorithm has reduced the simulation effort by a factor of approximately 10^{15}.

We are not aware of any different algorithms for the efficient solution of the SAT counting problem and as a result there are no benchmark results to which we can independently verify our new method. We can, however, put a deterministic lower bound on $|\mathscr{X}^*|$. The population $\mathcal{X}_T$ at the final iteration of Algorithm 3.1.1 is approximately uniformly distributed over the set $\mathscr{X}^*$ and as a result can be used to find some of the distinct solutions of the SAT problem. We ran Algorithm 3.1.1 10 times with $N = 10^4$ and were able to find 2253 distinct solutions amongst the 10 final populations generated at iteration T. We thus conclude that $|\mathscr{X}^*| \geqslant 2253$.

3.2 An Adaptive Generalized Splitting Algorithm

We now describe the algorithm which we use as a pilot run to estimate the splitting factors $\{\varrho_t\}$ and the levels $\{\gamma_t\}$. It is the earliest version of the GS algorithm [6], and we will refer to it as the ADAM algorithm, which stands for ADAptive Mulitilevel splitting algorithm. For example, Table 3.2 was created using Algorithm 3.2.1 with $N = 1000$, $\varrho = 0.5$, and the Markov transition density in Example 3.1.1.

Algorithm 3.2.1 (ADAM Algorithm) *Given the sample size N and the parameters $\varrho \in (0,1)$ and γ, execute the following steps.*

1. **Initialization.** *Set the counter $t = 1$ and execute:*

   - *Generate $\mathbf{X}_1, \ldots, \mathbf{X}_N \sim f(\mathbf{x})$ and denote $\mathcal{X}_0 = \{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$.*
   - *Let*
   $$\widetilde{\gamma}_t = \underset{\gamma \in \{S_1, \ldots, S_N\}}{\operatorname{argmin}} \left\{ \frac{1}{N} \sum_{i=1}^{N} I\{S(\mathbf{X}_i) \geqslant \gamma\} \leqslant \varrho \right\}, \quad \mathbf{X}_i \in \mathcal{X}_{t-1}, \tag{3.8}$$
   that is, $\widetilde{\gamma}_t$ is the smallest value from amongst $S(\mathbf{X}_1), \ldots, S(\mathbf{X}_N)$ such that $\frac{1}{N}\sum_{i=1}^{N} I\{S(\mathbf{X}_i) \geqslant \widetilde{\gamma}_t\} \leqslant \varrho$. Set $\gamma_t = \min\{\gamma, \widetilde{\gamma}_t\}$. Let $\mathcal{X}_t = \{\mathbf{X}_1, \ldots, \mathbf{X}_{N_t}\}$ be the largest subset of elements in $\mathcal{X}_{t-1}$ for which $S(\mathbf{X}) \geqslant \gamma_t$. Here, $N_t = |\mathcal{X}_t|$ is the number of elements in $\mathcal{X}_{t-1}$ for which $S(\mathbf{X}) \geqslant \gamma_t$. Then, $\varrho_t = \frac{N_t}{N}$ is an approximation of the probability $c_t = \mathbb{P}_f(S(\mathbf{X}) \geqslant \gamma_t \,|\, S(\mathbf{X}) \geqslant \gamma_{t-1})$, $\gamma_0 = -\infty$.

2. **Markov chain sampling.** *Same as Step 2 of Algorithm 3.1.1, except that in (3.2) the splitting factors are generated in a different way, namely,*
   $$\mathcal{S}_{ti} = \left\lfloor \frac{N}{N_t} \right\rfloor + B_i, \quad i = 1, \ldots, N_t.$$
   Here each $B_1, \ldots, B_{N_t}$ are $\mathsf{Ber}(1/2)$ random variables conditional on $\sum_{i=1}^{N_t} B_i = N \mod N_t$. More precisely, $(B_1, \ldots, B_{N_t})$ is a binary vector with joint pdf
   $$\mathbb{P}(B_1 = b_1, \ldots, B_{N_t} = b_{N_t}) = \frac{(N_t - r)!\, r!}{N_t!}\, I\{b_1 + \cdots + b_{N_t} = r\}, \quad b_i \in \{0,1\},$$
   where $r = N \mod N_t$. As a consequence of the different generation of the splitting factors, after resetting
   $$\mathcal{X}_t := \Big\{ \mathbf{Y}_{11}\,, \mathbf{Y}_{12}\,, \ldots, \mathbf{Y}_{1,\mathcal{S}_{t1}}\,, \cdots\cdots\cdots\,, \mathbf{Y}_{N_t 1}\,, \mathbf{Y}_{N_t 2}\,, \ldots, \mathbf{Y}_{N_t,\mathcal{S}_{tN_t}} \Big\},$$
   the set $\mathcal{X}_t$ contains exactly N elements.

3. **Updating and Estimation.** *Reset the counter* $t := t + 1$ *and proceed exactly as in the second bullet point of Step 1.*

4. **Stopping condition.** *If* $\gamma_t = \gamma$*, set* $T = t$ *and go to Step 5; otherwise, repeat from Step 2.*

5. **Final estimates.** *Deliver the estimated levels* $\gamma_1, \ldots, \gamma_T$*, the splitting factors* $\varrho_1, \ldots, \varrho_T$*, and the (biased) estimate of the rare-event probability:*

$$\widehat{\ell}_{\mathsf{ADAM}} = \prod_{t=1}^{T} \varrho_t = \frac{\prod_{t=1}^{T} N_t}{N^T}. \tag{3.9}$$

The main differences between the ADAM algorithm and the GS algorithm are the following. First, the difference in Step 2 of the ADAM algorithm is that the splitting factors are generated in a way that fixes the simulation effort at each stage to be N. Second, as seen from (3.8), the levels $\{\gamma_t\}$ are determined online using the random populations $\{\mathcal{X}_t\}$. As a consequence of these differences, the estimator $\widehat{\ell}_{\mathsf{ADAM}}$ is not unbiased and the algorithm does not provide a simple estimate for the variance of $\widehat{\ell}_{\mathsf{ADAM}}$.

Concerning the bias properties of the ADAM algorithm we have the following result [13].

Proposition 3.2.1 (Ceróu, Moral, Furon, Guyader) *Assume that the transition pdf* $\kappa_t(\mathbf{y} \,|\, \mathbf{x})$ *is independent of* $\mathbf{x}$ *for every* t*, that is, the Markov chain defining* κ_t *has made infinitely many moves at each stage. In addition, assume that* $S(\mathbf{x})$ *and* $f(\mathbf{x})$ *are continuous so that* (3.8) *implies* $\varrho_t = \varrho$ *for all* $t < T$*. Then, we have the following asymptotic expansion:*

$$\frac{\widehat{\ell}_{\mathsf{ADAM}}}{\ell} = 1 + \frac{1}{\sqrt{N}} \sqrt{(T-1)\frac{1-\varrho}{\varrho} + \frac{1-\varrho_T}{\varrho_T}}\, Z + \frac{(T-1)(1-\varrho)}{N\varrho} + o(N^{-1}),$$

where Z *has standard normal distribution.*

For a proof see [13]. Note that under the above assumptions $\widehat{\ell}_{\mathsf{ADAM}} = \varrho^{T-1}\varrho_T$ and $T = \lceil \log_\varrho(\ell) \rceil$.

Thus, the ADAM algorithm can be used as a stand-alone algorithm in the sense that it can provide an estimate of ℓ with negligible bias without the need for any preliminary simulation. For many medium scale problems we could not detect any substantive difference in the numerical performance of Algorithm 3.2.1 (ADAM) versus Algorithm 3.1.1 (GS). For example, for the same cost of 3.1 million samples ($N = 10^5$, $\varrho = 0.5$) Algorithm 3.2.1 gave an estimate of $\widehat{|\mathscr{X}^*|} = 2.26 \times 10^3$ with estimated relative error (RE) of 3%. This is an agreement with the same observation made in [13], namely that the bias of ADAM does not pose a significant problem. Despite this, we prefer to use the GS algorithm (with ADAM used for the pilot run) instead of using ADAM by itself, because the GS algorithm gives provably unbiased estimates for ℓ and the variance of $\widehat{\ell}$.

3.3 Fixed Effort Generalized Splitting

Our simulation experience is that when the splitting factors $\{\varrho_t\}$ and levels $\{\gamma_t\}$ are constructed using ADAM as the pilot algorithm, the GS algorithm rarely suffers from population explosions. There is a desire, however, to completely eliminate the theoretical possibility of an explosion, whilst retaining the unbiasedness of the estimator of ℓ. Recall that the GS algorithm is a generalization of the classical Fixed Splitting (FS) described in the introduction, and that the possibility of population explosions does not exist in the Fixed Effort (FE) splitting approach. Moreover, as discussed in the introduction, the FE splitting provides unbiased estimators. Thus, it is of interest to develop a generalized version of the FE splitting approach as well. Such a generalization results in the following algorithm, which will be shown to provide an unbiased estimator of ℓ.

Algorithm 3.3.1 (Fixed Effort Generalized Splitting) *Given a sequence $\{\gamma_t\}_{t=1}^T$ and sample size N, execute the following steps.*

1. Initialization. *Set the counter $t = 1$ and execute:*

- *Generate (not necessarily iid) $\mathbf{X}_1, \ldots, \mathbf{X}_N \sim f(\mathbf{x})$ and let $\mathcal{X}_0 = \{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$.*
- *Let $\mathcal{X}_t = \{\mathbf{X}_1, \ldots, \mathbf{X}_{N_t}\}$ be the largest subset of elements in $\mathcal{X}_{t-1}$ for which $S(\mathbf{X}) \geqslant \gamma_t$. Here, $N_t = |\mathcal{X}_t|$ is the random number of elements in $\mathcal{X}_{t-1}$ such that $S(\mathbf{X}) \geqslant \gamma_t$.*

2. Markov chain sampling. *For each $\mathbf{X}_i$ in $\mathcal{X}_t = \{\mathbf{X}_1, \ldots, \mathbf{X}_{N_t}\}$ sample independently:*

$$\mathbf{Y}_{ij} \sim \kappa_t(\mathbf{y} \,|\, \mathbf{X}_i), \quad j = 1, \ldots, \mathcal{S}_{ti}, \tag{3.10}$$

where

$$\mathcal{S}_{ti} = \left\lfloor \frac{N}{N_t} \right\rfloor + B_i, \quad i = 1, \ldots, N_t,$$

and $\kappa_t(\mathbf{y} \,|\, \mathbf{X}_i)$ is a Markov transition density with stationary pdf

$$f_t(\mathbf{y}) = \frac{f(\mathbf{y}) I\{S(\mathbf{y}) \geqslant \gamma_t\}}{\ell(\gamma_t)},$$

and each B_i is a $\mathsf{Ber}(1/2)$ random variable conditional on

$$\sum_{i=1}^{N_t} B_i = N \mod N_t.$$

Reset

$$\mathcal{X}_t := \Big\{ \mathbf{Y}_{11} \, , \mathbf{Y}_{12} \, , \ldots , \mathbf{Y}_{1,\mathcal{S}_{t1}} \, , \cdots\cdots\cdots \, , \, \mathbf{Y}_{N_t 1} \, , \mathbf{Y}_{N_t 2} \, , \ldots , \mathbf{Y}_{N_t,\mathcal{S}_{tN_t}} \Big\},$$

where $\mathcal{X}_t$ *contains* $|\mathcal{X}_t| = \sum_{i=1}^{N_t} \mathcal{S}_{ti} = N$ *elements.*

3. **Updating.** *Reset the counter* $t := t+1$ *and proceed exactly as in the second bullet point of Step 1.*

4. **Stopping condition.** *If* $t = T$ *or* $N_t = 0$*, set* $N_{t+1} = N_{t+2} = \cdots = N_T = 0$ *and go to Step 5; otherwise, repeat from Step 2.*

5. **Final estimator.** *Deliver the unbiased estimate of* ℓ*,* $\widehat{\ell}_{\mathsf{FE}} = \frac{1}{N^T} \prod_{t=1}^{T} N_t$.

We call Algorithm 3.3.1 the Fixed Effort Generalized Splitting (FE-GS) to distinguish it from the GS algorithm 3.1.1. The main difference between the FE-GS and GS algorithms is in Step 4, and in particular (3.10) and (3.2). In FE-GS the $\{\mathcal{S}_{ti}\}$ in (3.10) depend on the random variable N_t. Thus, the possibility of explosion is avoided by making the splitting dependent on the history of the process. The simulation effort at each stage t is fixed to be $\sum_{i=1}^{N_t} \mathcal{S}_{ti} = N$. Furthermore, the $\{\mathbf{Y}_{ij}\}$ are sampled by restarting the Markov transition density κ_t from the *same* point $\mathbf{X}_i$. In contrast, in the GS the $\{\mathcal{S}_{ti}\}$ in (3.2) are completely independent of $\{N_t\}$ and the past performance of the algorithm. In addition, each $\mathbf{Y}_{ij}$ is generated from $\kappa_t(\mathbf{y} \,|\, \mathbf{Y}_{i,j-1})$ instead of $\kappa_t(\mathbf{y} \,|\, \mathbf{X}_i)$, thus decreasing the dependence between, say, $\mathbf{Y}_{i1}$ and $\mathbf{Y}_{iN_t}$, and giving a more reliable estimate in Step 5. We mentioned in the introduction that in the ordinary FE splitting there is no easy way to estimate the variance of the estimator from a single simulation run. Similarly, there is no easy way to estimate the variance in the FE-GS algorithm. This is why, unlike Step 1 in Algorithm 3.1.1, Step 1 in Algorithm 3.3.1 does not require that the initial sample $\mathcal{X}_0$ be iid. Thus, an advantage of the GS algorithm is the availability of an estimate of the relative error from a single simulation run.

Figure 3.3 illustrates the typical evolution of the FE-GS algorithm in a two-dimensional state space. The entrance states at each stage are encircled. In contrast to Figure 3.1, the simulation effort at each stage is fixed at $N = 6$. Note how on Figure 3.3 the nature of the splitting is such that each point is always one Markov chain step away from an entrance state. As seen on Figure 3.3, we must generate $\{\mathbf{X}_{1j}\}$ by always starting the transition density κ_t from the entrance state $\mathbf{X}_1$, which decreases the diversity amongst $\{\mathbf{X}_{1j}\}$ — this is the price to pay for removing the possibility of a population explosion. Figure 3.4 shows the behavior of the score function $S(\mathbf{X})$, where $\mathbf{X}$ evolves as per the Markov chains on Figure 3.3. Time is measured in terms of the number of Markov chain steps.

Similar to the ordinary FE splitting, the FE-GS estimator is unbiased.

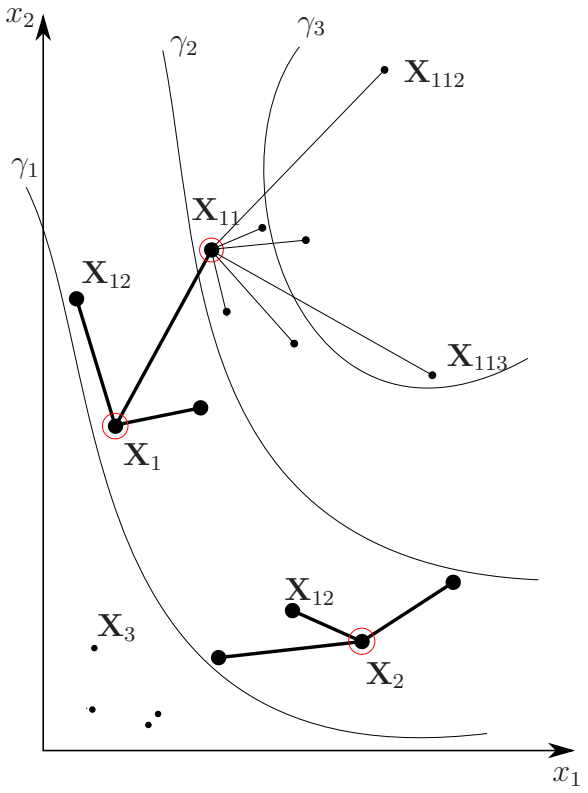

Figure 3.3: Typical evolution of the FE-GS algorithm in a two dimensional state space.

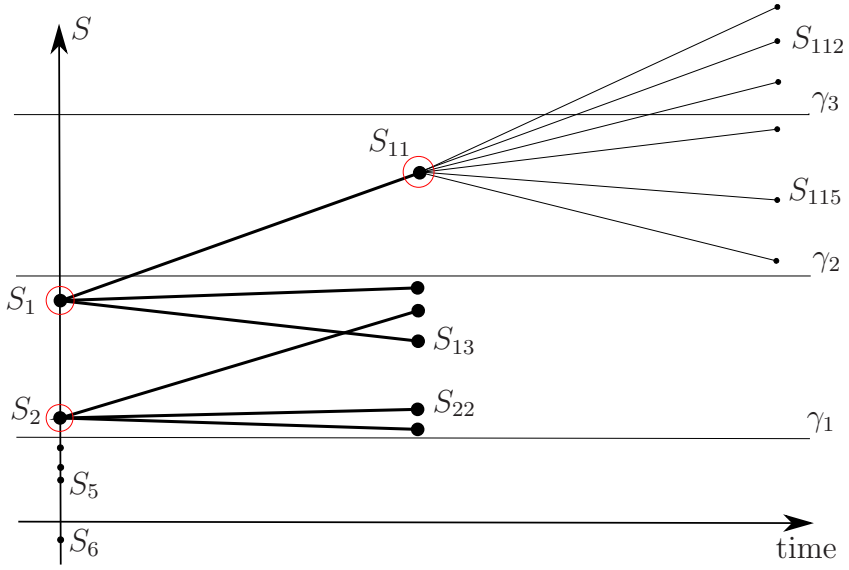

Figure 3.4: Typical evolution of $S(\mathbf{X})$ corresponding to the scenario in Figure 3.3.

Proposition 3.3.1 (Unbiasedness of the FE-GS estimator) *The estimator*

$$\widehat{\ell}_{\mathsf{FE}} = \prod_{t=1}^{T} \frac{N_t}{N} \quad \textit{is unbiased.} \tag{3.11}$$

Proof: Consider again for simplicity the case where $T = 3$. We can write

$$\begin{aligned}
N_1 &= \sum_{i=1}^{N} I_{\{S(\mathbf{X}_i)\geqslant\gamma_1\}}, \\
N_2 &= \sum_{i=1}^{N} I_{\{S(\mathbf{X}_i)\geqslant\gamma_1\}} \sum_{j=1}^{\mathcal{S}_{1i}} I_{\{S(\mathbf{X}_{ij})\geqslant\gamma_2\}}, \\
N_3 &= \sum_{i=1}^{N} I_{\{S(\mathbf{X}_i)\geqslant\gamma_1\}} \sum_{j=1}^{\mathcal{S}_{1i}} I_{\{S(\mathbf{X}_{ij})\geqslant\gamma_2\}} \sum_{k=1}^{\mathcal{S}_{2j}} I_{\{S(\mathbf{X}_{ijk})\geqslant\gamma_3\}},
\end{aligned}$$

where

$$\begin{aligned}
\mathbf{X}_i &\sim f(\,\cdot\,), \qquad i = 1,\ldots,N, \\
\mathbf{X}_{ij} &\sim \kappa_1(\,\cdot\,|\,\mathbf{X}_i), \quad i = 1,\ldots,N, \;\; j = 1,\ldots,\mathcal{S}_{1i}, \\
\mathbf{X}_{ijk} &\sim \kappa_2(\,\cdot\,|\,\mathbf{X}_{ij}), \quad \forall i,j, \quad k = 1,\ldots,\mathcal{S}_{2j},
\end{aligned}$$

independently. Then, $N_3N_2N_1 = \sum_{i=1}^{N} I_{\{S(\mathbf{X}_i)\geqslant\gamma_1\}} \; N_1 \sum_{j=1}^{\mathcal{S}_{1i}} I_{\{S(\mathbf{X}_{ij})\geqslant\gamma_2\}} \; N_2 \sum_{k=1}^{\mathcal{S}_{2j}} I_{\{S(\mathbf{X}_{ijk})\geqslant\gamma_3\}}$, from where it follows that:

$$\begin{aligned}
\mathbb{E}\left[\sum_{k=1}^{\mathcal{S}_{2j}} I_{\{S(\mathbf{X}_{ijk})\geqslant\gamma_3\}} \Big| \{\mathbf{X}_i, \mathbf{X}_{ij}, \mathcal{S}_{2j}\}\right] &= \sum_{k=1}^{\mathcal{S}_{2j}} \int \kappa_2(\mathbf{x}_{ijk}\,|\,\mathbf{x}_{ij}) I_{\{S(\mathbf{x}_{ijk})\geqslant\gamma_3\}}\, d\mathbf{x}_{ijk} \\
&= \mathcal{S}_{2j} \int \kappa_2(\mathbf{z}\,|\,\mathbf{x}_{ij}) I_{\{S(\mathbf{z})\geqslant\gamma_3\}}\, d\mathbf{z},
\end{aligned}$$

and (see Step 2. of Algorithm 3.3.1),

$$\mathbb{E}[\mathcal{S}_{ti}\,|\,N_t] = \left\lfloor \frac{N}{N_t} \right\rfloor + \mathbb{E}[B_i\,|\,N_t] = \left\lfloor \frac{N}{N_t} \right\rfloor + \frac{r}{N_t} = \frac{N}{N_t}.$$

Consequently,

$$\mathbb{E}\left[N_2 \sum_{k=1}^{\mathcal{S}_{2j}} I_{\{S(\mathbf{X}_{ijk})\geqslant\gamma_3\}} \Big| \{\mathbf{X}_i, \mathbf{X}_{ij}\}\right] = N\, \mathbb{E}_{\kappa_2(\,\cdot\,|\,\mathbf{X}_{ij})} I_{\{S(\mathbf{Z})\geqslant\gamma_3\}}.$$

As a result,

$$\mathbb{E}[N_3 N_2 N_1 \,|\, \{\mathbf{X}_i, \mathbf{X}_{ij}, \mathcal{S}_{1i}\}] = \sum_{i=1}^{N} I_{\{S(\mathbf{X}_i)\geqslant\gamma_1\}}\; N_1 \sum_{j=1}^{\mathcal{S}_{1i}} I_{\{S(\mathbf{X}_{ij})\geqslant\gamma_2\}}\; N\, \mathbb{E}_{\kappa_2(\,\cdot\,|\,\mathbf{X}_{ij})} I_{\{S(\mathbf{Z})\geqslant\gamma_3\}}$$

$$\mathbb{E}[N_3 N_2 N_1 \,|\, \{\mathbf{X}_i\}] = N^2 \sum_{i=1}^{N} I_{\{S(\mathbf{X}_i)\geqslant\gamma_1\}} \mathbb{E}_{\kappa_1(\,\cdot\,|\,\mathbf{X}_i)} \left[I_{\{S(\mathbf{Y})\geqslant\gamma_2\}}\; \mathbb{E}_{\kappa_2(\,\cdot\,|\,\mathbf{Y})} I_{\{S(\mathbf{Z})\geqslant\gamma_3\}} \right]$$

$$\mathbb{E}[N_3 N_2 N_1] = N^3\, \mathbb{E}_f \left[I_{\{S(\mathbf{X})\geqslant\gamma_1\}} \mathbb{E}_{\kappa_1(\,\cdot\,|\,\mathbf{X})} \left[I_{\{S(\mathbf{Y})\geqslant\gamma_2\}}\; \mathbb{E}_{\kappa_2(\,\cdot\,|\,\mathbf{Y})} I_{\{S(\mathbf{Z})\geqslant\gamma_3\}} \right] \right],$$

and

$$\frac{\mathbb{E}[N_3 N_2 N_1]}{N^3} = \iiint f(\mathbf{x}) \kappa_1(\mathbf{y}\,|\,\mathbf{x}) \kappa_2(\mathbf{z}\,|\,\mathbf{y}) I_{\{S(\mathbf{x})\geqslant\gamma_1\}} I_{\{S(\mathbf{y})\geqslant\gamma_2\}} I_{\{S(\mathbf{z})\geqslant\gamma_3\}}\, d\mathbf{z} d\mathbf{y} d\mathbf{x} = \ell,$$

where we have used the invariance property (3.7). We can thus deduce that (3.11) is unbiased. □

Example 3.3.1 (SAT counting continued) To illustrate the difference in performance between FE-GS and GS we consider the `RTI_k3_n100_m429_0` SAT instance ($n = 100$, $m = 429$). We ran the GS algorithm 3.1.1 with $N = 2 \times 10^4$, the Markov transition density in Example 3.1.1, and the levels and splitting factors in Table 3.3. We obtained an estimate of $\widehat{|\mathcal{X}^*|} = 2.15 \times 10^4$ with RE of 3%. The total cost was about 26 million samples, which is about 10^{-23} smaller than is required for the full enumeration of the sample space. In contrast, running the FE-GS algorithm 3.3.1 10 times with $N = 7 \times 10^4$ and using the levels given in Table 3.3, we obtained $\widehat{|\mathcal{X}^*|} = 2.3 \times 10^4$ with estimated RE of 23% (estimate from the 10 independent runs). Thus, GS outperforms FE-GS in this case. The explanation is that generating $\{\mathbf{Y}_{ij}\}$ by the Markov transition density $\kappa_t(\mathbf{y}\,|\,\mathbf{X}_i)$ with the same initial point $\mathbf{X}_i$ in (3.10) leads to higher correlation and less diversity in the population $\{\mathbf{Y}_{ij}\}$. We mention that 10 independent runs of ADAM algorithm with $N = 7 \times 10^4$ and $\varrho = 0.5$ gave $\widehat{|\mathcal{X}^*|} = 2.19 \times 10^4$ with estimated RE of 2.6% (estimate obtained from the 10 independent runs). The total cost is about 28 million samples and hence the performance of the ADAM algorithm in this case is comparable to the performance of the GS.

Simulation experience shows that neither the classical FE, nor the FS algorithms is universally superior, and selecting which algorithm is most appropriate for a given problem is not always obvious [24, 51]. Our simulation experience with the generalized versions of the classical FE and FS algorithms is similar. In particular, the relative advantages and disadvantages of Algorithms 3.1.1 (GS), 3.3.1 (FE-GS) and 3.2.1 (ADAM) are as follows.

1. The GS algorithm provides unbiased estimates for ℓ and its variance. However, the GS

Table 3.3: The sequence of levels and splitting factors used in Algorithm 3.1.1 for instance `RTI_k3_n100_m429_0`. The sequence was generated using the ADAM algorithm with $N = 10^3$ and $\varrho = 0.5$.

t	γ_t	ϱ_t	t	γ_t	ϱ_t
1	376	0.4940	20	411	0.3243
2	381	0.4534	21	412	0.3348
3	384	0.4994	22	413	0.3092
4	387	0.4104	23	414	0.2760
5	389	0.4916	24	415	0.2714
6	391	0.4563	25	416	0.2651
7	393	0.4285	26	417	0.2298
8	395	0.3767	27	418	0.229
9	397	0.3519	28	419	0.1922
10	399	0.2865	29	420	0.1630
11	401	0.2603	30	421	0.1432
12	403	0.2385	31	422	0.1321
13	404	0.4610	32	423	0.1113
14	405	0.4556	33	424	0.0826
15	406	0.4228	34	425	0.0661
16	407	0.4014	35	426	0.0328
17	408	0.3694	36	427	0.0299
18	409	0.3809	37	428	0.0116
19	410	0.3642	38	429	0.0009

requires a preliminary run to estimate the splitting factors $\{\varrho_t\}$ and the levels $\{\gamma_t\}$. If $\{\varrho_t\}$ are chosen badly, the population will either explode or die out before reaching the desired final level.

2. The FE-GS provides unbiased estimates for ℓ, but not for its variance. Although less efficient than the GS algorithm, the FE-GS algorithm does not require preliminary estimates of the splitting factors $\{\varrho_t\}$ and there is no possibility that the population will explode.

3. The ADAM algorithm does not provide unbiased estimates for either ℓ or its variance. However, it does not require any preliminary run, does not allow population explosions, and for many problems appears to be as accurate as the GS algorithm.

These conclusions are summarized in Table 3.4. In the next chapter we present various applications and extensions of the splitting methodology.

Table 3.4: Comparative advantages and disadvantages of the GS, FE-GS and ADAM algorithms.

	unbiased estimate of ℓ	unbiased estimate of $\mathrm{Var}(\widehat{\ell})$	no pilot run is needed to estimate $\{\varrho_t\}$	no pilot run is needed to estimate $\{\gamma_t\}$	explosions are prevented
GS	✓	✓	X	X	X
ADAM	X	X	✓	✓	✓
FE-GS	✓	X	✓	X	✓

Chapter 4

Extensions of the Splitting method

In this chapter we present the Generalized Splitting method for rare-event estimation.

4.1 Estimation of Integrals of the Form $\mathbb{E}_p[H(\mathbf{Z})]$

In the last section we showed how one can estimate rare-event probabilities of the form (1.2) using any of the ADAM, GS or FE-GS algorithms. In this section we extend the applicability of these algorithms to the more general problem of estimating integrals of the form (1.1). To this end we rewrite (1.1) using the following notation:

$$\mathcal{Z} = \mathbb{E}_p[H(\mathbf{Z})] = \int p(\mathbf{z})\, H(\mathbf{z})\, \mu(d\mathbf{z}), \tag{4.1}$$

so that now the aim is to estimate $\mathcal{Z}$. We show that each of GS or FE-GS algorithms can provide an unbiased estimate of $\mathcal{Z}$. First, note that

$$\mathbb{E}_p[H(\mathbf{Z})] = 2\,\mathbb{E}_p[H(\mathbf{Z})\, I_{\{H(\mathbf{Z})>0\}}] - \mathbb{E}_p[\,|H(\mathbf{Z})|\,],$$

so that without loss of generality we can consider estimating (4.1) for $H(\mathbf{z}) \geqslant 0$. Second, let $\widetilde{p}(\mathbf{z})$ be a proposal density from which it is easy to sample and which dominates $p(\mathbf{z})H(\mathbf{z})$ in the sense that

$$p(\mathbf{z})H(\mathbf{z}) \leqslant \mathrm{e}^{a\,\gamma+b}\,\widetilde{p}(\mathbf{z}), \qquad \forall \mathbf{z} \in \mathbb{R}^n, \tag{4.2}$$

for some $\gamma \geqslant (\ln(\mathcal{Z}) - b)/a$, where $a > 0$ and $b \in \mathbb{R}$ are fixed constants. Typically we have $\gamma \gg (\ln(\mathcal{Z}) - b)/a$ and in many cases it is natural to choose $\widetilde{p} \equiv p$. Next, in order to estimate $\mathcal{Z}$ we apply the following procedure.

Algorithm 4.1.1 (Estimation of $\mathcal{Z}$)

1. **Inputs.** *Suppose we are given a proposal $\widetilde{p}(\mathbf{z})$, parameters γ, a, b such that (4.2) holds, and algorithm A. Here A denotes one of the GS, FE-GS or ADAM algorithms.*

2. Estimation of ℓ. *Use algorithm* A *to estimate*

$$\ell(\gamma) = \mathbb{E}_f[I\{S(\mathbf{X}) \geqslant \gamma\}] = \int f(\mathbf{x})\, I\{S(\mathbf{x}) \geqslant \gamma\}\, d\mathbf{x}, \tag{4.3}$$

where the vector $\mathbf{x} = [\mathbf{z}, u]' \in \mathbb{R}^n \times [0,1]$ *augments* $\mathbf{z}$ *with the variable* $u \in [0,1]$*, the score* $S(\mathbf{x})$ *is given by*

$$S(\mathbf{x}) = \frac{1}{a} \ln\left(\frac{p(\mathbf{z})H(\mathbf{z})}{u\,\widetilde{p}(\mathbf{z})}\right) - \frac{b}{a},$$

and the density $f(\mathbf{x})$ *by*

$$f(\mathbf{x}) = \widetilde{p}(\mathbf{z}) \times I\{0 \leqslant u \leqslant 1\}, \quad \mathbf{x} \in \mathbb{R}^n \times \mathbb{R}.$$

3. Estimation of $\mathcal{Z}$. *An estimate of* $\mathcal{Z}$ *in* (4.1) *is:*

$$\widehat{\mathcal{Z}} = \mathrm{e}^{a\,\gamma+b}\, \widehat{\ell}(\gamma).$$

The following proposition shows that the estimate $\widehat{\mathcal{Z}}$ is unbiased if A is either the GS or the FE-GS algorithm.

Proposition 4.1.1 (Relation between ℓ and $\mathcal{Z}$) *The pdf*

$$\frac{p(\mathbf{z})H(\mathbf{z})}{\mathcal{Z}} \tag{4.4}$$

is the marginal density of $f_T(\mathbf{x}) = \frac{1}{\ell(\gamma_T)} f(\mathbf{x}) I\{S(\mathbf{x}) \geqslant \gamma_T\}$*, and* $\mathcal{Z} = \mathrm{e}^{a\,\gamma+b}\ell(\gamma)$*.*

Proof: Note that u is an *auxiliary variable* similar to the one used in the Accept-Reject method for random variable generation [66]. From (4.2) it follows that (with $\mathbf{x} = [\mathbf{z}, u]'$)

$$\begin{aligned}
\int_{\mathbb{R}} f_T(\mathbf{x})\, dx_{n+1} &= \int_{\mathbb{R}} \frac{\widetilde{p}(\mathbf{z}) I\{0 \leqslant u \leqslant 1\}}{\ell} I\left\{\frac{1}{a}\ln\left(\frac{p(\mathbf{z})H(\mathbf{z})}{u\widetilde{p}(\mathbf{z})}\right) - \frac{b}{a} \geqslant \gamma\right\} du \\
&= \frac{\widetilde{p}(\mathbf{z})}{\ell} \int_0^1 I\left\{u \leqslant \frac{p(\mathbf{z})H(\mathbf{z})}{\mathrm{e}^{a\gamma+b}\widetilde{p}(\mathbf{z})}\right\} du \\
&= \frac{p(\mathbf{z})H(\mathbf{z})}{\ell\ \mathrm{e}^{a\,\gamma+b}}\,,
\end{aligned}$$

because $\frac{p(\mathbf{z})H(\mathbf{z})}{\mathrm{e}^{a\gamma+b}\widetilde{p}(\mathbf{z})} \leqslant 1$ for all $\mathbf{z}$ by (4.2). Since $\mathcal{Z}$ is the normalizing constant of $p(\mathbf{z})H(\mathbf{z})$, we conclude that $\mathcal{Z}(\gamma) = \mathrm{e}^{a\,\gamma+b}\ell(\gamma)$. □

In practice $\mathrm{e}^{a\,\gamma+b} \gg \mathcal{Z}$, and hence $\ell = \mathcal{Z}\,\mathrm{e}^{-a\,\gamma-b}$ is a rare-event probability. To illustrate the efficiency of Algorithm 4.1.1 in estimating (4.1), we consider two examples.

Example 4.1.1 The following toy example is adapted from [68]. Consider the problem of estimating without bias the normalizing constant $\mathcal{Z}$ of the pdf proportional to

$$h(\mathbf{z}) = \exp\left(-\frac{z_1^2 + z_2^2 + (z_1 z_2)^2 - 2\lambda z_1 z_2}{2}\right), \quad \mathbf{z} \in \mathbb{R}^2,$$

for some parameter $\lambda \in \mathbb{R}$. An approximate value $\mathcal{Z} \approx 3.5390 \times 10^{26}$ was obtained using the deterministic recursive Simpson's rule [22]. The density $h(\mathbf{z})/\mathcal{Z}$ is plotted on Figure 4.1.1 for $\lambda = 12$.

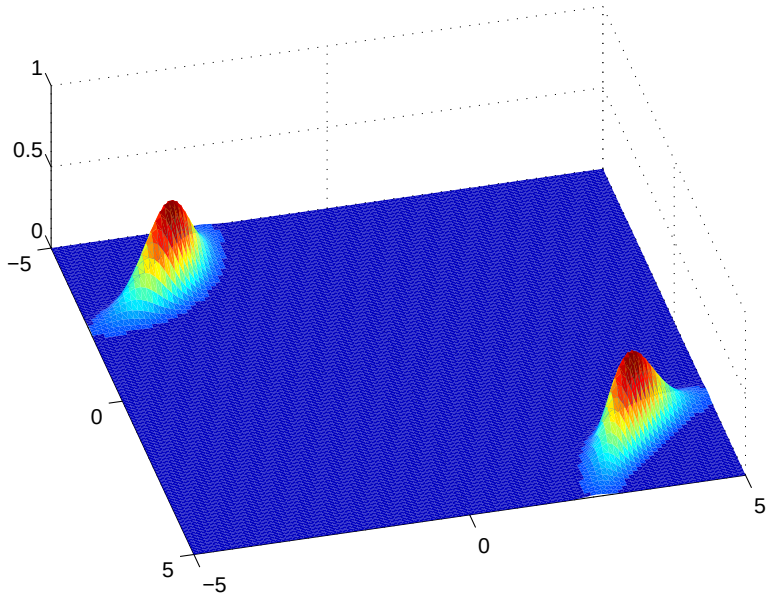

Figure 4.1: Plot of the pdf $h(\mathbf{z})/\mathcal{Z}$ for $\lambda = 12$.

This is a problem of the form (4.1) with $p(\mathbf{z}) = \frac{1}{2\pi}\exp\left(-\frac{z_1^2+z_2^2}{2}\right)$ and $H(\mathbf{z}) = 2\pi\exp\left(-\frac{(z_1 z_2)^2 - 2\lambda z_1 z_2}{2}\right)$. Let $\widetilde{p}(\mathbf{z}) = p(\mathbf{z})$, $a = \frac{1}{2}$, $b = \frac{\lambda^2}{2} + \ln(2\pi)$, and A be the GS algorithm in Step 1 of Algorithm 4.1.1. Then, (4.3) can be written as

$$\ell(\gamma) = \mathbb{P}_f\left(-(Z_1 Z_2 - \lambda)^2 - 2\ln(U) \geqslant \gamma\right),$$

where the vector $\mathbf{x} = [\mathbf{z}, u]'$ is augmented such that

$$f(\mathbf{x}) = \frac{1}{2\pi}\exp\left(-\frac{z_1^2 + z_2^2}{2}\right) \times I\{0 < u < 1\},$$

and the level $\gamma = 0$ is such that (4.2) holds. To apply the GS algorithm for the estimation of

(4.3), we need to specify the transition pdf κ_t with stationary density

$$f_t(\mathbf{x}) = \frac{f(\mathbf{x})\, I\left\{-(z_1 z_2 - \lambda)^2 - 2\ln(u) \geqslant \gamma_t\right\}}{\ell(\gamma_t)}.$$

A move from $\mathbf{X}$ to $\mathbf{X}^*$ using the transition density $\kappa_t(\mathbf{X}^* \,|\, \mathbf{X})$ consists of the following (systematic) Gibbs sampling procedure.

Algorithm 4.1.2 (Defining the transition density $\kappa_t(\mathbf{X}^* \,|\, \mathbf{X})$)

1. Given a state $\mathbf{X} = [\mathbf{Z}, U]'$ for which $S(\mathbf{X}) \geqslant \gamma_t$, generate $Z_1^* \sim f_t(z_1^* \,|\, Z_2, U)$; that is, draw Z_1^* from a truncated standard normal density on the interval $[I_1, I_2]$, where $I_1 = \min\{\frac{\lambda-\mu}{Z_2}, \frac{\lambda+\mu}{Z_2}\}$, $I_2 = \max\{\frac{\lambda-\mu}{Z_2}, \frac{\lambda+\mu}{Z_2}\}$, and $\mu = \sqrt{-\gamma_t - 2\ln(U)}$.

2. Generate $Z_2^* \sim f_t(z_2^* \,|\, Z_1^*, U)$; that is, draw Z_2^* from a truncated standard normal density on the interval $[I_1, I_2]$, where $I_1 = \min\{\frac{\lambda-\mu}{Z_1^*}, \frac{\lambda+\mu}{Z_1^*}\}$, $I_2 = \max\{\frac{\lambda-\mu}{Z_1^*}, \frac{\lambda+\mu}{Z_1^*}\}$, and $\mu = \sqrt{-\gamma_t - 2\ln(U)}$.

3. Generate a uniform random variable U^* on the interval $[0, \mu]$, where

$$\mu = \min\left\{1,\ \exp\left(-\frac{\gamma_t + (Z_1^* Z_2^* - \lambda)^2}{2}\right)\right\}.$$

   Output $\mathbf{X}^* = [Z_1^*, Z_2^*, U^*]$.

To estimate ℓ we applied the GS algorithm with $N = 2000$, $\lambda = 12$, and the levels and splitting factors in Table 4.1.

Table 4.1: Levels and splitting factors used to compute the normalizing constant of $h(\mathbf{z})$.

t	1	2	3	4	5	6
γ_t	-117.91	-77.03	-44.78	-20.18	-4.40	0
ϱ_t	0.1	0.1	0.1	0.1	0.1	0.2853

We obtained $\widehat{\ell} = 2.92 \times 10^{-6}$ with relative error of 5%. Hence, in Step 3 of Algorithm 4.1.1 we obtain $\widehat{\mathcal{Z}} = \widehat{\ell} \times \mathrm{e}^{a\,\gamma+b} = \widehat{\ell} \times 2\pi \mathrm{e}^{\lambda^2/2} = 3.41 \times 10^{26}$ with a relative error of 5%. Note that Table 4.1 was computed using the ADAM algorithm with $\varrho = 0.1$, $N = 2000$ and using the same transition density κ_t. The combined simulation effort of the GS algorithm and the ADAM algorithm was about 1.2×10^5 samples. In contrast, CMC estimation of $\mathcal{Z}$ via the estimator

$\frac{1}{M}\sum_{i=1}^{M} H(\mathbf{Z}_i),\ \{\mathbf{Z}_i\}_{\text{iid}} \sim p(\mathbf{z}),\ M = 1.2\times 10^5$ gave an estimate of 1.6×10^{26} with relative error of 60%.

For this two-dimensional example we were able to verify the simulation results using deterministic quadrature. The constant $\mathcal{Z}$ in the next example, however, cannot be easily computed using an alternative method due to the high-dimensionality of the problem.

Example 4.1.2 (Rosenbrock function) The following example is adapted from [68]. Consider computing the normalizing constant of the pdf proportional to

$$h(\mathbf{z}) = \exp\left(-\lambda R(\mathbf{z})\right), \quad z_i \in [-2,2],\ i = 1,\ldots,n,$$

where

$$R(\mathbf{z}) = \sum_{i=1}^{n-1} 100(z_{i+1} - z_i^2)^2 + (z_i - 1)^2$$

is the Rosenbrock function in $\mathbb{R}^n$, [67]. Again, the problem is of the form (4.1), with $p(\mathbf{z}) = 1/4^n,\ \mathbf{z}\in[-2,2]^n$ and $H(\mathbf{z}) = 4^n h(\mathbf{z})$. Let $\widetilde{p}(\mathbf{z}) = p(\mathbf{z})$, $a = \lambda$ and $b = \ln(4^n)$. Then, (4.3) can be written as

$$\ell(\gamma) = \mathbb{P}_f\left(-\frac{\ln(U)}{\lambda} - R(\mathbf{Z}) \geqslant \gamma\right),$$

where

$$f(\mathbf{x}) = \frac{\prod_{i=1}^{n} I\{-2 \leqslant z_i \leqslant 2\}}{4^n} \times I\{0 < u < 1\}, \quad \mathbf{x} = [\mathbf{z}, u]',$$

and $\gamma = 0$ is such that (4.2) is a tight bound. To estimate ℓ we apply the ADAM algorithm using a transition density $\kappa_t(\mathbf{x}^* \,|\, \mathbf{x})$ with stationary pdf

$$f_t(\mathbf{x}) = \frac{f(\mathbf{x}) I\left\{-\frac{\ln(u)}{\lambda} - R(\mathbf{z}) \geqslant \gamma_t\right\}}{\ell(\gamma_t)}.$$

A move from $\mathbf{X}$ to $\mathbf{X}^*$ consists of the following Gibbs sampling procedure.

1. Given a state $\mathbf{X} = [\mathbf{Z}, U]'$ such that $S(\mathbf{X}) \geqslant \gamma_t$, set $\Sigma = R(\mathbf{Z})$. Generate $U^* \sim f_t(u \,|\, \mathbf{Z})$; that is, draw a uniform random variable U^* on the interval $[0, \mu]$, where

$$\mu = \min\left\{1, \exp\left(-\lambda\gamma_t - \lambda\Sigma\right)\right\}.$$

2. Reset

$$\Sigma := \Sigma - 100(Z_2 - Z_1^2)^2 - (Z_1 - 1)^2 + \frac{\ln(U^*)}{\lambda}.$$

Generate $Z_1^* \sim f_t(z_1 \,|\, U^*, Z_2, \ldots, Z_n)$; that is, Z_1^* is a uniform random variable on the set $\{[r_1, r_2] \cup [r_3, r_4]\} \cap [-2, 2]$, where $r_1 < r_2,\ r_3 < r_4$ are the real roots of the quartic

equation $a_1x^4 + a_2x^3 + a_3x^2 + a_4x + a_5 = 0$ with coefficients:

$$\begin{aligned} a_1 &= 100 \\ a_2 &= 0 \\ a_3 &= 1 - 200Z_2^2 \\ a_4 &= -2 \\ a_5 &= 1 + 100Z_2^2 + \gamma_t + \Sigma\ . \end{aligned}$$

Depending on the coefficients, the quartic equation has either 2 or 4 real roots. Thus in some cases $r_3 = r_1$ and $r_4 = r_2$ and Z_1^* will be a uniform random variable on the set $[r_1, r_2] \cap [-2, 2]$.

3. For each $j = 2, \ldots, n-1$, reset

$$\begin{aligned} \Sigma := \Sigma &- 100(Z_{j+1} - Z_j^2)^2 - (Z_j - 1)^2 - 100(Z_j^2 - 2(Z_{j-1}^*)^2 Z_j) \\ &+ 100(Z_2 - (Z_1^*)^2)^2 + (Z_1^* - 1)^2. \end{aligned}$$

Generate $Z_j^* \sim f_t(z_j \,|\, U^*, Z_1^*, \ldots, Z_{j-1}^*, Z_{j+1}, \ldots, Z_n)$; that is, Z_j^* is a uniform random variable on the set $\{[r_1, r_2] \cup [r_3, r_4]\} \cap [-2, 2]$, where $r_1 < r_2,\ r_3 < r_4$ are the real roots of the quartic equation $a_1x^4 + a_2x^3 + a_3x^2 + a_4x + a_5 = 0$ with coefficients:

$$\begin{aligned} a_1 &= 100 \\ a_2 &= 0 \\ a_3 &= 101 - 200Z_{j+1} \\ a_4 &= -2 - 200(Z_{j-1}^*)^2 \\ a_5 &= 1 + 100Z_{j+1}^2 + \gamma_t + \Sigma\ . \end{aligned}$$

4. Reset

$$\begin{aligned} \Sigma := \Sigma &- 100(Z_n - (Z_{n-1}^*)^2)^2 \\ &+ 100(Z_{j+1} - (Z_j^*)^2)^2 + (Z_j^* - 1)^2 + 100((Z_j^*)^2 - 2(Z_{j-1}^*)^2 Z_j^*). \end{aligned}$$

Generate $Z_n^* \sim f_t(z_n \,|\, U^*, Z_1^*, \ldots, Z_{n-1}^*)$; that is, Z_n^* is a uniform random variable on the set $[r_1, r_2] \cap [-2, 2]$, where $r_1 < r_2$ are the roots of the quadratic equation $a_1x^2 + a_2x + a_3 = 0$

with coefficients:

$$
\begin{aligned}
a_1 &= 100 \\
a_2 &= -200(Z^*_{n-1})^2 \\
a_3 &= 100(Z^*_{n-1})^4 + \gamma_t + \Sigma \,.
\end{aligned}
$$

As a numerical example, consider the case where $\lambda = 10^4$ and $n = 10$. We ran the ADAM algorithm 400 independent times with $\varrho = 0.5$ and $N = 1000$, and obtained $\widehat{\ell} = 9.7 \times 10^{-36}$ with estimated relative error (using the data from the 400 runs) of 7%. Therefore, $\widehat{\mathcal{Z}} = \widehat{\ell} \times \mathrm{e}^{a\gamma+b} = \widehat{\ell} \times 4^{10} \approx 1.0 \times 10^{-29}$ with relative error of 7%. Each run of the ADAM algorithm took about 117 iterations ($T = 117$), giving a total simulation effort of $N \times 400 \times T = 46.8 \times 10^6$ samples. For the same simulation effort the CMC estimator $\frac{1}{M}\sum_{i=1}^{M} \exp(-\lambda R(\mathbf{Z}))$, $M = 46.8 \times 10^6$, with $\{\mathbf{Z}_i\}$ independent and uniformly distributed on $[-2, 2]^{10}$, did not give meaningful results (relative error of 99.9%). In fact, to achieve a relative error of 7% using CMC estimation would require a simulation effort of approximately 2×10^{37} samples.

Numerical minimization of the Rosenbrock function $R(\mathbf{z})$ is a challenging minimization problem [67]. It is commonly used as a test case for a wide range of numerical optimization routines. The function $R(\mathbf{z})$ has a global minimum of 0 at $\mathbf{z} = [1, \ldots, 1]'$. One way in which $R(\mathbf{z})$ could be minimized is to sample approximately from the Boltzmann density $\mathrm{e}^{-\lambda R(\mathbf{z})}/\mathcal{Z}$, $\mathbf{z} \in [-2, 2]^n$ for a large value of λ. In Example 4.1.2, as a consequence of estimating the constant $\mathcal{Z}$ using Algorithm 4.1.1, we also obtain an estimate for the global minimizer of $R(\mathbf{z})$. In particular, the population $\mathcal{X}_T = \{\mathbf{X}_1, \ldots, \mathbf{X}_{N_T}\}$ at the final iteration of the ADAM algorithm is approximately distributed according to the stationary density $f_T(\mathbf{x})$. Hence, $\mathbf{Z}_i$ in $\mathbf{X}_i = [\mathbf{Z}_i, U_i]'$ is approximately distributed according to the marginal (Boltzmann density) $p(\mathbf{z})H(\mathbf{z})/\mathcal{Z} = \mathrm{e}^{-\lambda R(\mathbf{z})}/\mathcal{Z}$, and we can use $\mathbf{Z}_i : \; i = \mathrm{argmin}_j \, R(\mathbf{Z}_j)$ as an estimate for the global minimizer of $R(\mathbf{z})$. For the numerical example considered above we obtained

$$
\mathbf{Z}_i = [1.00,\ 0.99,\ 0.99,\ 0.99,\ 1.00,\ 1.00,\ 1.00,\ 1.00,\ 1.00,\ 1.00]^T
$$

with $R(\mathbf{Z}_i) \approx 5 \times 10^{-5}$. Thus, the result is close to the true minimizer $[1, \ldots, 1]^T$. Therefore, we can use Algorithm 4.1.1 (with A set to be ADAM algorithm) as an optimization algorithm. This is similar to the *Simulated Annealing* algorithm [54, 68], in which the MH sampler is used to approximately sample from the Boltzmann density and minimize the function $R(\mathbf{z})$. In the next section we use the ADAM algorithm as an optimization routine. In particular we look at difficult combinatorial optimization examples.

4.2 Combinatorial Optimization via the ADAM

Combinatorial optimization has always been an important and challenging part of optimization theory. A well-known instance of a difficult combinatorial optimization problem is the 0-1 knapsack problem, defined as:

$$\begin{aligned} &\max_{\mathbf{x}} \sum_{j=1}^{n} p_j x_j, \qquad x_i \in \{0,1\}, \\ &\text{subject to : } \sum_{j=1}^{n} w_{ij} x_j \leqslant c_i, \quad i = 1, \ldots, m. \end{aligned} \tag{4.5}$$

Here $\{p_i\}$ and $\{w_{ij}\}$ are positive weights and $\{c_i\}$ are positive cost parameters. To make (4.5) easier to handle as an estimation problem, we note that (4.5) is equivalent to $\max_{\mathbf{x}\in\{0,1\}^n} S(\mathbf{x})$, where $\mathbf{x} = (x_1, \ldots, x_n)$ is a binary vector and

$$S(\mathbf{x}) = C(\mathbf{x}) + \sum_{j=1}^{n} p_j x_j = \alpha \sum_{i=1}^{m} \min\left\{ c_i - \sum_{j=1}^{n} w_{ij} x_j, 0 \right\} + \sum_{j=1}^{n} p_j x_j,$$

with $\alpha = (1 + \sum_{j=1}^{n} p_j) / \max_{i,j}\{c_i - w_{ij}\}$. Note that the constant α is such that if $\mathbf{x}$ satisfies all the constraints in (4.5), then $C(\mathbf{x}) = 0$ and $S(\mathbf{x}) = \sum_{j=1}^{n} p_j x_j \geqslant 0$. Alternatively, if $\mathbf{x}$ does not satisfy all of the constraints in (4.5), then $C(\mathbf{x}) \leqslant -(1 + \sum_{j=1}^{n} p_j x_j)$ and $S(\mathbf{x}) \leqslant -1$. To this optimization problem we can associate the problem of estimating the rare-event probability

$$\ell(\gamma) = \mathbb{P}_f \left(S(\mathbf{X}) \geqslant \gamma \right), \quad \gamma \in \left[0, \sum_{j=1}^{n} p_j \right],$$

where $f(\mathbf{x}) = \frac{1}{2^n}$ for all $x_i \in \{0,1\}$, that is, $\mathbf{X}$ is a vector of independent Bernoulli random variables with success probability 1/2. An important difference in the optimization setting is that we are not interested *per se* in obtaining an unbiased estimate for the rare-event probability. Rather we only wish to approximately sample from the pdf $f_t(\mathbf{x}) = f(\mathbf{x}) I\{S(\mathbf{x}) \geqslant \gamma_t\} / \ell(\gamma_t)$ for as large a value of γ_t as the algorithm finds possible. Given this objective we run the ADAM algorithm with $\gamma = \infty$ and modify Steps 4 and 5 in Algorithm 3.2.1 in the following way.

4. **Stopping Condition.** *If there is no progress in increasing γ_t over a number of iterations, that is, if $\gamma_t = \gamma_{t-1} = \cdots = \gamma_{t-d}$ for some user-specified positive integer d, set $T = t$ and go to Step 5; otherwise, repeat from Step 2.*

5. Final estimates. *Deliver the vector* $\mathbf{x}^*$ *from the set*

$$\mathcal{X}_T = \{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$$

for which $S(\mathbf{X}_i)$ *is maximal as an estimate for the global maximizer of* (4.5).

The action of the transition density $\kappa_t(\mathbf{y} \,|\, \mathbf{x})$ is equivalent to the following Gibbs sampling procedure:

1. Given a state $\mathbf{x}$ such that $S(\mathbf{x}) \geqslant \gamma_t$, generate $Y_1 \sim f_t(y_1 \,|\, x_2, \ldots, x_n)$;

2. For each $k = 2, \ldots, n-1$, generate $Y_k \sim f_t(y_k \,|\, Y_1, \ldots, Y_{k-1}, x_{k+1}, \ldots, x_n)$;

3. Finally, generate $Y_n \sim f_t(y_n \,|\, Y_1, \ldots, Y_{n-1})$.

Here the conditional density is given by

$$f_t(y_k \,|\, \mathbf{y}_{-k}) \propto I\left\{C(\mathbf{y}) + p_k y_k \geqslant \gamma_t - \sum_{j \neq k} p_j y_j\right\},$$

where $\mathbf{y}_{-k}$ denotes the vector $\mathbf{y}$ with the k-th element removed. Sampling a random variable Y_k from such a conditional can be accomplished as follows. Draw $B \sim \mathsf{Ber}(1/2)$, if $S([y_1, \ldots, y_{k-1}, B, y_{k+1}, \ldots, y_n]) \geqslant \gamma_t$, then set $Y_k = B$, otherwise set $Y_k = 1 - B$.

As a particular example, consider the *Sento1.dat* knapsack problem given in

`http://people.brunel.ac.uk/mastjjb/jeb/orlib/files/mknap2.txt`

The problem has 30 constraints and 60 variables. We selected $\varrho = 0.01$ and $N = 10^3$, and the algorithm was stopped after no progress was observed ($d = 1$). We ran the algorithm ten independent times. The algorithm always found the optimal solution. A typical evolution of the algorithm is given in Table 4.2. The total sample size used is 10^4 which is about 8.67×10^{-15} of the total effort needed for the complete enumeration of the 2^{60} possible binary vectors.

Table 4.2: Typical evolution of Algorithm 3.2.1 as an optimization routine for knapsack problem. The maximum value for this problem is 6704.

t	γ_t	t	γ_t
1	-5708.55	6	6051.74
2	1296.94	7	6346.32
3	3498.60	8	6674.92
4	4567.91	9	6704
5	5503.22	10	6704

Traveling Salesman Problem

The objective of the TSP is to find the shortest tour in a complete weighted graph containing n nodes. The problem can be formulated as minimizing the cost function

$$S(\mathbf{x}) = \sum_{i=1}^{n-1} C_{x_i,x_{i+1}} + C_{x_n,x_1},$$

where $\mathbf{x} = (x_1, \ldots, x_n)$ is a permutation of $(1, \ldots, n)$, x_i is the $i-$th node (or city) to be visited in a tour represented by $\mathbf{x}$, and C_{ij} is a cost (or distance associated with the edges of the graph) from node (or city) i to node j. We now discuss how to apply the ADAM algorithm to find the optimal tour. The sequence of distributions from which we wish to approximately sample is

$$f_t(\mathbf{x}) = \frac{f(\mathbf{x}) I\{-S(\mathbf{x}) \geqslant \gamma_t\}}{\ell(\gamma_t)}, \quad \mathbf{x} \in \mathscr{X}, \quad t = 1, 2, \ldots,$$

where $f(\mathbf{x})$ is the uniform density over the set $\mathscr{X}$ of all possible tours (i.e., the set of all possible permutations of $(1, \ldots, n)$), and $\{\gamma_t\}$ is an increasing sequence of levels. To apply Algorithm 3.2.1 (with Steps 4 and 5 modified as in the knapsack example in the beginning of this section), set $\gamma = \infty$ and specify the transition pdf $\kappa_t(\,\cdot \mid \cdot\,)$ in Step 2 via the following conditional sampling. Given a tour $\mathbf{x}$, update $\mathbf{x}$ to $\mathbf{y}$ with probability $I\{-S(\mathbf{y}) \geqslant \gamma_t\}$, where $\mathbf{y}$ has the tour between the x_i-th and x_j-th cities ($j \neq i$) reversed. Repeat b number of times. For example, if $n = 10$ and $\mathbf{x} = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$ and $i = 4, j = 9$, then we accept the state $\mathbf{y} = (1, 2, 3, 9, 8, 7, 6, 5, 4, 10)$ with probability $I\{-S(\mathbf{y}) \geqslant \gamma_t\}$; otherwise we do not update $\mathbf{x}$. The conditional sampling is therefore similar to the *2-opt* heuristic commonly employed in

conjunction with simulated annealing [68].

In the course of the conditional sampling the score function is updated as follows ($j > i$):

$$S(\tilde{\mathbf{x}}) = S(\mathbf{x}) - C_{x_{i-1},x_i} - C_{x_j,x_{j+1}} + C_{x_{i-1},x_j} + C_{x_i,x_{j+1}}.$$

We now present a number of numerical experiments which demonstrate the performance of the algorithm. Table 4.3 summarizes the results from a number of benchmark problems from the TSP library:

`http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/tsp/`

The experiments were executed using Matlab 7 on a laptop PC with a 2GHz Intel Centrino Duo CPU.

Table 4.3: Case studies for the symmetric TSP. The experiments were repeated ten times and the average minimum and maximum of $S(\mathbf{x})$ were recorded. The parameters of the ADAM algorithm are $\varrho = 0.5$, $N = 10^2$ and the 2-opt updating is repeated $b = n \times 50$ times, where n is the size of the problem. The CPU times are in seconds.

file	$-\gamma_T$	min	mean	max	CPU	$\bar{T}$
burma14	3323	3323	3323	3323	3.5	45.8
ulysses16	6859	6859	6859	6859	4.7	53.2
ulysses22	7013	7013	7013	7013	9.9	79.7
bayg29	1610	1610	1610	1610	16	113
bays29	2020	2020	2020	2020	16	110.7
dantzig42	699	699	699	699	38	188.2
eil51	426	426	427.6	430	57	249.1
berlin52	7542	7542	7542	7542	60	232.5
st70	675	675	675.8	680	116	370.6
eil76	538	538	543.9	547	145	428.6
pr76	108159	108159	108216	108304	130	372.3

Observe that the algorithm finds the optimal solution in all cases out of ten trials. In some cases, the algorithm found the optimal solution ten out of ten times. The number of

iterations necessary to solve a problem increases with the size of the problem and is roughly equal to $\log_\varrho(n!)$, which is bounded from above by $n \log_\varrho(n)$. Table 4.4 gives some medium-scale examples. Again note the good performance of the algorithm in finding the optimal tour.

Table 4.4: Medium-scale case studies for the symmetric TSP. The experiments were repeated ten times and the average minimum and maximum of $S(\mathbf{x})$ were recorded. The parameters of the ADAM algorithm are $\varrho = 0.5$, $N = 10^2$ and the 2-opt updating is repeated $b = n \times 50$ times, where n is the size of the problem. The CPU times are in seconds.

file	$-\gamma_T$	min	mean	max	CPU	$\bar{T}$
a280	2579	2581	2594.4	2633	5763	2026.7
ch130	6110	6110	6125.9	6172	1096	742.8
eil101	629	643	647.6	654	703	620.4
gr120	6942	6956	6969.2	6991	935	668.4
gr137	69853	69853	69911.8	70121	1235	781
kroA100	21282	21282	21311.2	21379	634	527.2
kroB100	22141	22141	22201	22330	634	526.7
kroC100	20749	20749	20774.4	20880	636	528.8
kroD100	21294	21294	21295.5	21309	630	529
kroE100	22068	22068	22123.1	22160	650	526.5
lin105	14379	14379	14385.6	14401	712	561.2
pr107	44303	44303	44305.3	44326	780	569.3
pr124	59030	59030	59048.4	59076	1018	691.5
pr136	96772	97102	97278.2	97477	1180	760
pr144	58537	58537	58640.9	59364	1406	827.6
pr152	73682	73682	73833	74035	1620	886.6
rat99	1211	1211	1213.2	1218	614	561.5
rd100	7910	7910	7910.8	7916	622	534.7
si175	21407	21013	21027.2	21051	2030	1066.3
st70	675	676	677.6	681	297	369.3
swiss42	1273	1273	1273	1273	103	180
u159	42080	42080	42383	42509	1688	934.4

Quadratic Assignment Problem

The quadratic assignment problem (QAP) is one of the most challenging problems in optimization theory. It has various applications, such as computer chip design, optimal resource allocation, and scheduling. In the context of optimal allocation, the objective is to find an assignment of a set of n facilities to a set of n locations such that the total cost of the assignment is minimized. The QAP can be formulated as the problem of minimizing the cost function

$$S(\mathbf{x}) = \sum_{i=1}^{n} \sum_{j=1}^{n} F_{ij}\, D_{x_i,x_j},$$

where $\mathbf{x} = (x_1, \ldots, x_n)$ is a permutation on $(1, \ldots, n)$, F is an $n \times n$ *flow* matrix, that is, F_{ij} represents the flow of materials from facility i to facility j and D is an $n \times n$ distance matrix, such that D_{ij} is the distance between location i and location j. We assume that both F and D are symmetric matrices. We now apply Algorithm 3.2.1 (modifying steps 4. and 5. as in the knapsack example in the beginning of this section) to the QAP with $\gamma = \infty$. The transition density $\kappa_t(\mathbf{y} \,|\, \mathbf{x})$ with stationary pdf $f_t(\mathbf{y}) = f(\mathbf{y}) I\{-S(\mathbf{y}) \geqslant \gamma_t\}/\ell(\gamma_t)$ (here $f(\mathbf{y})$ is again the uniform density over all permutations) is specified by the following conditional sampling procedure.

1. Given the permutation $\mathbf{x} = (x_1, \ldots, x_n)$, draw a pair of indices (I, J) such that $I \neq J$ and both I and J are uniformly distributed over the integers $1, \ldots, n$.

2. Given $(I, J) = (i, j)$, generate the pair (Y_i, Y_j) from the conditional bivariate pdf

$$f_t(y_i, y_j \,|\, \mathbf{x}_{-i,-j}), \quad (y_i, y_j) \in \{(x_i, x_j), (x_j, x_i)\},$$

   where $\mathbf{x}_{-i,-j}$ is the same as $\mathbf{x}$, except that the i-th and j-th elements are removed.

3. Set $\mathbf{x} = [x_1, x_2, \ldots, y_i, x_{i+1}, \ldots, x_{j-1}, y_j, \ldots, x_{n-1}, x_n]$, assuming $j > i$.

4. Repeat steps 1 through 3 above b times.

Sampling from $f_t(y_i, y_j \,|\, \mathbf{x}_{-i,-j})$ is accomplished as follows. Given the state $\mathbf{x}$, we update $\mathbf{x}$ to $\mathbf{y}$ with probability $I\{-S(\mathbf{y}) \geqslant \gamma_t\}$, where $\mathbf{y}$ is identical to $\mathbf{x}$ except that the i-th and j-th positions are exchanged. For example, if $\mathbf{x} = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$ and $i = 3, j = 7$, then $\mathbf{y} = (1, 2, 7, 4, 5, 6, 3, 8, 9, 10)$. In the course of the conditional sampling, the score function is updated as follows:

$$S(\mathbf{y}) = S(\mathbf{x}) + 2 \sum_{k \neq i,j} (F_{kj} - F_{ki})(D_{x_k,x_i} - D_{x_k,x_j}).$$

We now present a number of numerical experiments which demonstrate the performance of the algorithm. Table 4.5 summarizes the results from a number of benchmark problems from the TSP library:

`http://www.seas.upenn.edu/qaplib/inst.html`

Table 4.5: Case studies for the symmetric QAP. The experiments were repeated ten times and the average, minimum, and maximum of $S(\mathbf{x})$ were recorded. The parameters of the ADAM algorithm are $\varrho = 0.5$, $N = 10^3$ and for the conditional sampling $b = n$, where n is the size of the problem.

`file`	$-\gamma_T$	min	mean	max	CPU	$\bar{T}$
`chr12a.dat`	9552	9552	9552	9552	21	45.2
`chr12b.dat`	9742	9742	9742	9742	21	45.4
`chr12c.dat`	11156	11156	11159	11186	20	42.5
`chr15a.dat`	9896	9896	9942.8	10070	36	53
`chr15b.dat`	7990	7990	8100	8210	36	53.4
`chr15c.dat`	9504	9504	10039	10954	36	53.4
`chr18a.dat`	11098	11098	11102.4	11142	60	64
`chr18b.dat`	1534	1534	1534	1534	54	57.3
`chr20a.dat`	2192	2192	2344	2406	75	66.9
`chr20b.dat`	2298	2352	2457.8	2496	70	64.5
`chr20c.dat`	14142	14142	14476.8	14812	85	77.7
`chr22a.dat`	6156	6156	6208.6	6298	105	81.4
`chr22b.dat`	6194	6194	6290.4	6362	97	75.3
`chr25a.dat`	3796	3796	4095.6	4286	147	90.1

The algorithm, although quite slower, works for large scale asymmetric QAP instances as well. For the instance `Lipa90b.dat` of size 90, we obtained the optimal solution tour ($S(\mathbf{x}^*) = 12490441$) ten out of ten times with algorithmic parameters $N = 10^2$, $\varrho = 0.5$, $b = 900$. The average iterations for convergence (using the same stopping criterion as above) was 505 and the average CPU time was 10426 seconds.

4.3 Sensitivity Analysis

A common problem in the study of *discrete-event systems* [68] is the estimation of functions of the form:

$$\mathcal{Z}(\boldsymbol{\lambda}) = \int p(\mathbf{z}) H(\mathbf{z}) W(\mathbf{z}; \boldsymbol{\lambda}) \mu(d\mathbf{z}), \tag{4.6}$$

where W is a likelihood ratio that depends on a *sensitivity* parameter $\boldsymbol{\lambda} \in \mathbb{R}^d$. The objective is to estimate $\mathcal{Z}(\boldsymbol{\lambda})$ as a function of $\boldsymbol{\lambda}$ from a single simulation run. Usually one is interested in estimating $\mathcal{Z}(\boldsymbol{\lambda})$ in the neighborhood of $\widetilde{\boldsymbol{\lambda}}$, where $W(\mathbf{z}; \widetilde{\boldsymbol{\lambda}}) = 1, \ \forall \mathbf{z}$. Such problems can be easily handled by the generalized splitting algorithms after some minor modifications. To estimate the function $\mathcal{Z}(\boldsymbol{\lambda})$ we follow exactly the same steps as given in Algorithm 4.1.1, except that (4.3) is replaced with

$$\ell_{\boldsymbol{\lambda}} = \ell_{\boldsymbol{\lambda}}(\gamma) = \mathbb{E}_f[W_{\boldsymbol{\lambda}}(\mathbf{X}) \ I\{S(\mathbf{X}) \geqslant \gamma\}], \quad W_{\boldsymbol{\lambda}}(\mathbf{X}) = W(\mathbf{Z}; \boldsymbol{\lambda}),$$

and the input algorithm A is modified so that it provides an estimate of $\ell_{\boldsymbol{\lambda}}(\gamma)$ instead of (4.3). Suppose that A is the GS algorithm, then the modified version differs only in Step 5 and reads as follows.

Algorithm 4.3.1 (Splitting algorithm for estimating $\ell_{\boldsymbol{\lambda}} = \mathbb{E}_f W_{\boldsymbol{\lambda}}(\mathbf{X}) I\{S(\mathbf{X}) \geqslant \gamma\}$)
Given a sequence $\{\gamma_t, \varrho_t\}_{t=1}^T$ and a sample size N, execute the following steps.

1-4. *Steps 1 through 4 are the same as in Algorithm 3.1.1.*

5. Final estimator. *Deliver the unbiased estimate of $\ell_{\boldsymbol{\lambda}}$:*

$$\widehat{\ell}_{\boldsymbol{\lambda}} = \frac{\prod_{t=1}^T \varrho_t}{N_0} \sum_{i=1}^{N_T} W_{\boldsymbol{\lambda}}(\mathbf{X}_i),$$

where each $\mathbf{X}_i \in \mathcal{X}_T$. An unbiased estimate of the variance is

$$\widehat{\mathrm{Var}(\widehat{\ell}_{\boldsymbol{\lambda}})} = \frac{\prod_{t=1}^T \varrho_t^2}{N_0(N_0 - \varrho_1)} \sum_{j=1}^{N_0/\varrho_1} \left(O_{\boldsymbol{\lambda}}^{(j)} - \frac{\varrho_1}{N_0} \sum_{i=1}^{N_T} W_{\boldsymbol{\lambda}}(\mathbf{X}_i) \right)^2, \tag{4.7}$$

where

$$O_{\boldsymbol{\lambda}}^{(j)} = \sum_{i=1}^{N_T} I_j(\mathbf{X}_i) W_{\boldsymbol{\lambda}}(\mathbf{X}_i),$$

with $I_j(\mathbf{X}_i) = 1$ if $\mathbf{X}_i$ is a Markov chain move that shares a common history with the j-th point in the initial population $\mathcal{X}_0$, and $I_j(\mathbf{X}_i) = 0$ otherwise.

We note the following aspect of the algorithm. First, since $W_{\widetilde{\boldsymbol{\lambda}}}(\mathbf{X}_i) = 1$ by definition, we have $\widehat{\ell}_{\widetilde{\boldsymbol{\lambda}}} = \frac{\prod_{t=1}^{T} \varrho_t}{N_0}$ and $\widehat{\ell}_{\widetilde{\boldsymbol{\lambda}}}$ is simply the point estimator of (4.1). We can thus write

$$\widehat{\ell}_{\boldsymbol{\lambda}} = \widehat{\ell}_{\widetilde{\boldsymbol{\lambda}}} \, \frac{1}{N_T} \sum_{i=1}^{N_T} W_{\boldsymbol{\lambda}}(\mathbf{X}_i).$$

Second, $\sum_{j=1}^{N_0/\varrho_1} I_j(\mathbf{X}_i) = 1$ for all $\mathbf{X}_i$ and

$$\sum_{j=1}^{N_0/\varrho_1} O_{\boldsymbol{\lambda}}^{(j)} = \sum_{i=1}^{N_T} W_{\boldsymbol{\lambda}}(\mathbf{X}_i).$$

For example, on Figure 3.1 we have

$$O_{\boldsymbol{\lambda}}^{(1)} = \sum_{k=3}^{10} W_{\boldsymbol{\lambda}}(\mathbf{X}_{12k}) + \sum_{k \in \{7,8,10\}} W_{\boldsymbol{\lambda}}(\mathbf{X}_{16k})$$

and $O_{\boldsymbol{\lambda}}^{(j)} = 0$ for $j = 2, \ldots, 10$, because points $\mathbf{X}_{12k}$, $k = 3, \ldots, 10$ and $\mathbf{X}_{16k}$, $k = 7, 8, 10$ comprise the set $\mathcal{X}_T$ and they all belong to a branch sprouting from the first point in $\mathcal{X}_0$.

Finally, the FE-GS and ADAM are similarly modified to provide an estimate of $\ell_{\boldsymbol{\lambda}}$. The determination of the levels and splitting factors is the same as applied to the estimation of $\ell_{\widetilde{\boldsymbol{\lambda}}} = \mathbb{E}_f I\{S(\mathbf{X}) \geqslant \gamma_t\}$. Typically, the estimation of $\ell_{\boldsymbol{\lambda}}$ is accurate in the neighborhood of $\widetilde{\boldsymbol{\lambda}}$. We next apply Algorithm 4.3.1 to the estimation of the partition function in the context of the well-known Ising model [54].

Example 4.3.1 (Ising Model) Consider the Ising model on a lattice with n sites. Each site i is in a state $z_i \in \{-1, 1\}$. The interactions between sites are encoded by an $n \times n$ matrix A such that $A_{ij} = 0$ if sites i and j do not interact or $i = j$. We wish to estimate the partition function $\mathcal{Z}(\lambda)$ of the Boltzmann pdf proportional to

$$h(\mathbf{z}) = \exp\left(\lambda \, \mathbf{z} \, A \, \mathbf{z}'\right), \quad \mathbf{z} = [z_1, \ldots, z_n] \in \{-1, 1\}^n.$$

We are interested in estimation of $\mathcal{Z}(\lambda)$ in the neighborhood of $\widetilde{\lambda} = 2$ (see [54]). This problem is of the form (4.6) with $p(\mathbf{z}) = 1/2^n$, $\mathbf{z} \in \{-1, 1\}^n$,

$$H(\mathbf{z}) = 2^n \exp\left(2 \, \mathbf{z} \, A \, \mathbf{z}'\right),$$

and

$$W(\mathbf{z}; \lambda) = \exp\left((\lambda - 2) \, \mathbf{z} \, A \, \mathbf{z}'\right).$$

Let $\widetilde{p}(\mathbf{z}) = p(\mathbf{z}),\ a = 1,\ b = \ln(2^n)$, and A be Algorithm 4.3.1. Then, $\ell_\lambda(\gamma_t)$ can be written as:

$$\ell_\lambda(\gamma_t) = \mathbb{E}_f[W(\mathbf{Z};\lambda)\ I\{2\ \mathbf{Z}\,A\,\mathbf{Z}' - \ln(U) \geqslant \gamma_t\}],$$

where $(\mathbf{x} = [\mathbf{z}, u]')$

$$f(\mathbf{x}) = \frac{1}{2^n}\ I\{0 < u < 1\}, \quad \mathbf{z} \in \{0,1\}^n,$$

and

$$\gamma = \gamma_T = 2\sum_{j=1}^{n}\sum_{j=1}^{n}|A_{ij}|,$$

so that (4.2) is satisfied. The transition pdf $\kappa_t(\mathbf{x}^* \,|\, \mathbf{x})$ has stationary density

$$f_t(\mathbf{x}) = \frac{f(\mathbf{x})I\{2\ \mathbf{z}\,A\,\mathbf{z}' - \ln(u) \geqslant \gamma_t\}}{\ell(\gamma_t)}$$

A move from $\mathbf{X}$ to $\mathbf{X}^*$ consists of the following Gibbs sampling procedure.

Algorithm 4.3.2 (Gibbs sampling for Ising model)

1. Given a state $\mathbf{X} = [\mathbf{Z}, U]'$ such that $S(\mathbf{X}) \geqslant \gamma_t$, generate $U^* \sim f_t(u \,|\, \mathbf{Z})$; that is, draw a uniform random variable U^* on the interval $[0, \mu]$, where

$$\mu = \min\left\{1, \exp\left(2\ \mathbf{Z}\,A\,\mathbf{Z}' - \gamma_t\right)\right\}.$$

2. For $k = 1, \ldots, n$, generate $Z_k^* \sim f_t(z_k \,|\, U^*, Z_1^*, \ldots, Z_{k-1}^*, Z_{k+1}, \ldots, Z_n)$; that is, generate V uniformly on $\{-1, 1\}$, and if $Z_k \neq V$ and

$$2\,\mathbf{V}\,A\,\mathbf{V}' - \ln(U^*) \geqslant \gamma_t, \quad \mathbf{V} = [Z_1^*, \ldots, Z_{k-1}^*, V, Z_k, \ldots, Z_n],$$

   set $Z_k^* = V$; otherwise, set $Z_k^* = Z_k$.

As a numerical example, consider the case where $\mathbf{z}\,A\,\mathbf{z}' = \sum_{i=1}^{n-1} z_i z_{i+1},\ n = 50$, and $\bar{\lambda} = 2$, then $\mathcal{Z}(\lambda) = \ell_\lambda\,\mathrm{e}^\gamma 2^n$ and $\gamma_T = \gamma = 98$. For this case it is known that $\mathcal{Z}(\lambda) = 2(\mathrm{e}^\lambda + \mathrm{e}^{-\lambda})^{n-1}$, see [54]. Figure 4.2 compares the exact $\mathcal{Z}(\lambda)$ (solid) versus the estimated $\widehat{\mathcal{Z}}(\lambda)$ (dashed), and indicates the relative error of the estimate, computed using the variance estimator (4.7). The estimate was computed using Algorithm 4.3.1 with $N = 10^4$, and the levels and splitting factors given in Table 4.6. The total simulation cost was about 8×10^6 samples. Note that the estimate has low relative error for $\lambda > \bar{\lambda} = 2$, but that it deviates from the true value and yields a large estimated relative error for $\lambda < 1$. This behavior is consistent with the *trust region* of the likelihood ratio in sensitivity analysis [69].

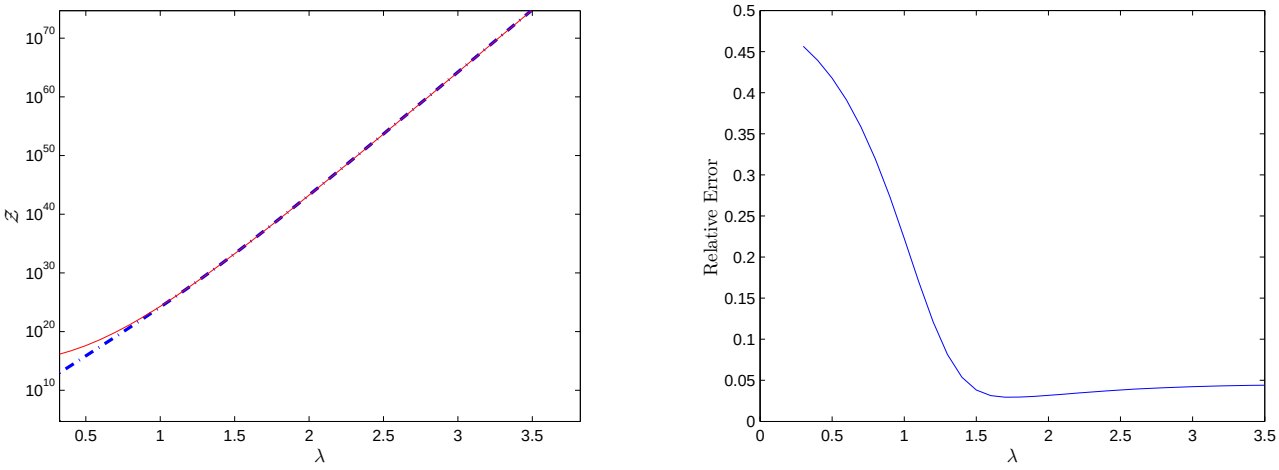

Figure 4.2: Left: Plot of the true $\mathcal{Z}(\lambda)$ (solid) versus the estimate $\widehat{\mathcal{Z}}(\lambda)$ (dashed). Right: Plot of the estimated relative error, computed using (4.7).

Table 4.6: The levels and splitting factors were computed using the ADAM algorithm with $N = 10^4$ and $\varrho = 0.01$.

t	1	2	3	4	5	6	7	8
γ_t	34.282	51.608	66.204	75.48	83.856	91.091	96.688	98
ϱ_t	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.3387

4.4 MCMC Sampling

In this section we consider using the splitting algorithms as an alternative to standard MCMC sampling from multidimensional pdfs of the form (3.1). Note that since (4.4) can be viewed as a marginal density of (3.1), sampling from (4.4) reduces to the problem of sampling from (3.1). We show that the population $\mathcal{X}_T$ in the final stage of the splitting algorithms can be treated as a sample from the multidimensional pdf (3.1) even in cases where standard MCMC algorithms are impractical due to poor mixing. In addition, we also provide a convergence diagnostic which tests the hypothesis that the population $\mathcal{X}_T$ is drawn from the target pdf (3.1). Deciding when a Markov chain has converged is an important problem in applications of MCMC. Many methods for diagnosing convergence have been proposed, ranging from simple graphical methods to computationally intensive hypothesis tests [9, 11].

For clarity of presentation we explain how to sample from (3.1) in a separate algorithm called the Fixed-Effort-sampler (FE-sampler).

Algorithm 4.4.1 (FE-sampler) *Given a sequence* $\{\gamma_t, \varrho_t\}_{t=1}^T$, *set* $s_t = \lceil \varrho_{t+1}^{-1} \rceil$, $\forall t < T$ *and execute the following steps.*

1. **Initialize.** *Set the counter* $t = 1$. *Keep generating* $\mathbf{X} \sim f(\mathbf{x})$, *until* $S(\mathbf{X}) \geqslant \gamma_1$. *Let* $\mathbf{X}^1 = \mathbf{X}$ *be the output. Note that* $\mathbf{X}^1$ *has density* $f(\mathbf{x}) I\{S(\mathbf{x}) \geqslant \gamma_1\}/c_1$.

2. **Markov chain sampling.** *Generate*

$$\mathbf{Y}_j \sim_{iid} \kappa_t(\mathbf{y} \,|\, \mathbf{X}^t), \quad j = 1, \ldots, s_t, \tag{4.8}$$

where $\kappa_t(\mathbf{y} \,|\, \mathbf{X}^t)$ *is a reversible Markov transition density with stationary pdf* $f_t(\mathbf{y})$. *Let*

$$N_{t+1} = \sum_{j=1}^{s_t} I\{S(\mathbf{Y}_j) \geqslant \gamma_{t+1}\}.$$

If $N_{t+1} = 0$, *repeat from Step 1, otherwise continue with Step 3.*

3. **Updating.** *Let* $\mathbf{X}^{t+1}$ *be a randomly selected point from the set of points* $\{\mathbf{Y}_1, \ldots, \mathbf{Y}_{s_t}\}$ *such that* $S(\mathbf{X}^{t+1}) \geqslant \gamma_{t+1}$. *The pdf of* $\mathbf{X}^{t+1}$ *is thus given by*

$$\frac{I\{S(\mathbf{x}) \geqslant \gamma_{t+1}\} \kappa_t(\mathbf{x} \,|\, \mathbf{X}^t)}{c_{t+1}(\mathbf{X}^t)}, \tag{4.9}$$

where $c_{t+1}(\mathbf{y}) = \int I\{S(\mathbf{x}) \geqslant \gamma_{t+1}\} \kappa_t(\mathbf{x} \,|\, \mathbf{y})\, d\mathbf{x}$ *is the probability that a move of the Markov chain starting in state* $\mathbf{y}$ *has score above* γ_{t+1}. *Note that an unbiased estimate of* $c_{t+1}(\mathbf{X}^t)$ *is*

$$\widehat{c}_{t+1}(\mathbf{X}^t) = \frac{N_{t+1}}{s_t},$$

so that $\mathbb{E}[\widehat{c}_{t+1}(\mathbf{X}^t) \,|\, \mathbf{X}^t] = c_{t+1}(\mathbf{X}^t)$. *Reset the counter* $t := t + 1$.

4. **Final Output.** *If* $t = T$, *output* $\{\widehat{c}_{t+1}(\mathbf{X}^t)\}_{t=1}^{T-1}$ *and* $\vec{\mathbf{X}} = (\mathbf{X}^1, \ldots, \mathbf{X}^T)$, *otherwise repeat from Step 2.*

Comparing (4.8) with (3.10) we see that the FE-sampler is conceptually the same as the FE-GS algorithm, except that here the kernel κ_t has to be reversible. The proposed diagnostic test is based on the following result.

Proposition 4.4.1 *The final state* $\mathbf{X}^T$ *of the FE-sampler has pdf*

$$f_T(\mathbf{x}) = \frac{I\{S(\mathbf{x}) \geqslant \gamma\} f(\mathbf{x})}{\ell(\gamma)}$$

if and only if $\sum_{t=1}^{T-1} \ln(c_{t+1}(\mathbf{x}^t)) = const.$ *for every* $\vec{\mathbf{x}} = (\mathbf{x}^1, \ldots, \mathbf{x}^T)$ *such that* $S(\mathbf{x}^t) \geqslant \gamma_t$, $t = 1 \ldots, T-1$. *In other words, the Markov chain of the FE-sampler is in stationarity if and only if* $\sum_{t=1}^{T-1} \ln(c_{t+1}(\mathbf{x}^t))$ *does not depend on* $\vec{\mathbf{x}}$.

Proof: First, the joint pdf of $\vec{\mathbf{X}}$ is:

$$\widehat{f}_T(\vec{\mathbf{x}}) = \frac{f(\mathbf{x}^1)\, I\{S(\mathbf{x}^1) \geqslant \gamma_1\}}{c_1} \prod_{t=1}^{T-1} \frac{I\{S(\mathbf{x}^{t+1}) \geqslant \gamma_{t+1}\} \kappa_t(\mathbf{x}^{t+1} \,|\, \mathbf{x}^t)}{c_{t+1}(\mathbf{x}^t)}.$$

Using the reversibility of the transition densities $\{\kappa_t\}$, we can write the joint pdf as

$$\widehat{f}_T(\vec{\mathbf{x}}) = \frac{f(\mathbf{x}^T) I\{S(\mathbf{x}^T) \geqslant \gamma_T\}}{c_1} \prod_{t=1}^{T-1} \frac{\kappa_t(\mathbf{x}^t \,|\, \mathbf{x}^{t+1})}{c_{t+1}(\mathbf{x}^t)}, \tag{4.10}$$

Ideally we would like the joint pdf in (4.10) to be identical to the target:

$$\begin{aligned} f_T(\vec{\mathbf{x}}) &:= \frac{f(\mathbf{x}^T) I\{S(\mathbf{x}^T) \geqslant \gamma_T\}}{c_1} \prod_{t=1}^{T-1} \frac{\kappa_t(\mathbf{x}^t \,|\, \mathbf{x}^{t+1})}{c_{t+1}} \\ &= \frac{f(\mathbf{x}^T)\, I\{S(\mathbf{x}^T) \geqslant \gamma_T\}}{\ell} \prod_{t=1}^{T-1} \kappa_t(\mathbf{x}^t \,|\, \mathbf{x}^{t+1}), \end{aligned} \tag{4.11}$$

because then $\mathbf{x}^T$ has the desired marginal density $f_T(\mathbf{x})$. We can measure how close the sampling density $\widehat{f}_T(\vec{\mathbf{x}})$ is from the target density $f_T(\vec{\mathbf{x}})$ using any distance measure from the Csisár's *ϕ-divergence* family of measures [8, 68]. A convenient member of Csisár's family of measures is the χ^2 *goodness of fit* divergence defined as

$$\mathcal{D}(p \to q) = \frac{1}{2} \int \frac{[p(\mathbf{x}) - q(\mathbf{x})]^2}{p(\mathbf{x})}\, d\mathbf{x} = -\frac{1}{2} + \frac{1}{2} \mathbb{E}_p \frac{q^2(\mathbf{X})}{p^2(\mathbf{X})},$$

for any given pair of pdfs p and q. Thus, we can measure the closeness between the sampling pdf (4.10) and the target pdf (4.11) via

$$\mathcal{D}(\widehat{f}_T \to f_T) = -\frac{1}{2} + \frac{1}{2} \mathbb{E}_{\widehat{f}_T} \prod_{t=1}^{T-1} \frac{c_{t+1}^2(\mathbf{X}^t)}{c_{t+1}^2}.$$

Hence, after rearranging, we have

$$2\,\frac{\ell^2}{c_1^2}\,\mathcal{D}(\widehat{f}_T \to f_T) = \mathbb{E}_{\widehat{f}_T} \prod_{t=1}^{T-1} c_{t+1}(\mathbf{X}^t) - \frac{\ell^2}{c_1^2} = \mathrm{Var}_{\widehat{f}_T} \prod_{t=1}^{T-1} c_{t+1}(\mathbf{X}^t),$$

where we have used the fact that $\mathbb{E}_{\widehat{f}_T} \prod_{t=1}^{T-1} c_{t+1}(\mathbf{X}^t) = \ell/c_1$. It follows that the distance between (4.10) and (4.11) is zero if and only if $\mathrm{Var}_{\widehat{f}_T} \prod_{t=1}^{T-1} c_{t+1}(\mathbf{X}^t) = 0$. In other words, $\prod_{t=1}^{T-1} c_{t+1}(\mathbf{X}^t) = const.$ or $\sum_{t=1}^{T-1} \ln(c_{t+1}(\mathbf{X}^t)) = const.$ for any $\vec{\mathbf{X}} \sim \widehat{f}_T(\vec{\mathbf{x}})$. This completes the proof. □

To test if $\sum_{t=1}^{T-1} \ln(c_{t+1}(\mathbf{X}^t)) = const.$ let $\vec{\mathbf{X}}_1, \vec{\mathbf{X}}_2, \ldots, \vec{\mathbf{X}}_M \sim \widehat{f}_T(\vec{\mathbf{x}})$ be a population from the sampling density (4.10), and let C be the following $(T-1) \times M$ matrix of estimates

$$C = \begin{bmatrix} \ln(\widehat{c}_2(\mathbf{X}_1^1)) & \ln(\widehat{c}_3(\mathbf{X}_1^2)) & \cdots & \ln(\widehat{c}_T(\mathbf{X}_1^{T-1})) \\ \ln(\widehat{c}_2(\mathbf{X}_2^1)) & \ln(\widehat{c}_3(\mathbf{X}_2^2)) & \cdots & \ln(\widehat{c}_T(\mathbf{X}_2^{T-1})) \\ \ln(\widehat{c}_2(\mathbf{X}_3^1)) & \ln(\widehat{c}_3(\mathbf{X}_3^2)) & \cdots & \ln(\widehat{c}_T(\mathbf{X}_3^{T-1})) \\ \vdots & \vdots & \cdots & \vdots \\ \ln(\widehat{c}_2(\mathbf{X}_M^1)) & \ln(\widehat{c}_3(\mathbf{X}_M^2)) & \cdots & \ln(\widehat{c}_T(\mathbf{X}_M^{T-1})) \end{bmatrix},$$

where the i-th row depends on $\vec{\mathbf{X}}_i = (\mathbf{X}_i^1, \mathbf{X}_i^2, \ldots, \mathbf{X}_i^T)$. If the chain of the FE-sampler samples according to the target, then the sums across each row of matrix C should be roughly the same. Hence, given a large enough M, we can conduct a two-way ANOVA test for row effects in matrix C. Each column of C represents a different level, while each row of C represents a given factor. We use the following χ^2 test to diagnose convergence of the Markov chains. Compute the following statistics:

$$\text{row means:} \quad \bar{C}_{i\bullet} = \frac{1}{T-1} \sum_{j=1}^{T-1} C_{ij},$$

$$\text{column means:} \quad \bar{C}_{\bullet j} = \frac{1}{M} \sum_{i=1}^{M} C_{ij},$$

$$\text{overall mean:} \quad \bar{C} = \frac{1}{T-1} \sum_{j=1}^{T-1} \bar{C}_{\bullet j},$$

$$\text{“row effect” sum of squares:} \quad \mathrm{SSTR} = \sum_{i=1}^{M} (\bar{C}_{i\bullet} - \bar{C})^2,$$

$$\text{“within row” variance:} \quad \mathrm{SSE} = \frac{1}{(M-1)(T-1)^2} \sum_{j=1}^{T-1} \sum_{i=1}^{M} (C_{ij} - \bar{C}_{\bullet j})^2,$$

then the test statistic $\mathcal{T} = \frac{\text{SSTR}}{\text{SSE}}$ has approximately χ^2 distribution with $M-1$ degrees of freedom. The p-value is $1-\Phi(\mathcal{T})$, where Φ is the cdf of the χ^2 distribution with $M-1$ degrees of freedom. Alternatively, one can use the Friedman's robust nonparametric ANOVA test [35] to detect row dependencies in matrix C. Note that Friedmann's test uses only the ranks of the data and will thus give the same p-value when applied to the matrix with entries $\{\widehat{c}_{j+1}(\mathbf{X}_i^j)\}$ as applied to matrix C with entries $\{\ln(\widehat{c}_{j+1}(\mathbf{X}_i^j))\}$.

Remark 4.4.1 The only reason for presenting the FE-sampler as an algorithm distinct from the FE-GS algorithm is so that we can introduce a new notation that helps in the theoretical analysis of the sampling mechanism within the FE-GS algorithm. In practice, we use the output generated from the FE-GS algorithm itself and never run the FE-sampler as a distinct algorithm. All the information required for the construction of matrix C is generated during the running of the FE-GS algorithm. Hence, once we obtain an estimate of the quantity of interest (4.6), the χ^2 convergence diagnostic comes at no extra computational cost.

Remark 4.4.2 Similar convergence diagnostics, using multiple independent Markov chains, have been investigated in [10, 26]. In particular, the Gelman-Rubin diagnostic [26] uses multiple independent chains and compares the variance within a single chain with the variance across the multiple chains using a χ^2 statistic. The differences between the proposed diagnostic and diagnostics such as the Gelman-Rubin test can be summarized as follows.

First, the Gelman-Rubin diagnostic requires the estimation of an initial overly-dispersed estimate of the target distribution. This is achieved by locating all the modes of the target density and the fitting of a parametric mixture model with as many components as there are modes. The mixture model is used to generate the initializing states of the Markov chains. In contrast, our convergence diagnostic is completely independent of the initializing states of the Markov chains and does not require preliminary knowledge of the modes of the target density.

Second, the Gelman-Rubin diagnostic has to be reapplied for every particular quantity that can be estimated using the chains' output [66]. In contrast, the proposed diagnostic requires only the computation of the quantities $\{c_{t+1}(\mathbf{X}^t)\}$, and it tests for a sufficient condition for convergence. The sufficiency is due to the fact that the distance between the sampling pdf (4.10) and the target pdf (4.11) is zero if and only if the sum $\sum_{t=1}^{T-1} \ln(c_{t+1}(\mathbf{X}^t))$ does not depend on $\vec{\mathbf{X}}$. The sufficiency implies that the diagnostic is not only useful for detecting any transient behavior of the FE-sampler, but for testing the stationarity of the FE-sampler as well.

Finally, there is no need to test the *convergence of averages* (which is guaranteed by the ergodic theorem) [66], because the FE-sampler already provides an unbiased estimate of (4.1) and (4.6).

Remark 4.4.3 In the paper [7], we incorrectly claimed that a precursor of the FE-sampler generates exact samples from a given target density. As the results in this section show, the FE-sampler does not yield exact sampling from the target pdf. It samples approximately from the target, just like existing MCMC samplers.

Since for estimation purposes we use the GS algorithm (Algorithm 3.1.1) more often than the FE-GS algorithm (Algorithm 3.3.1), it is desirable to use the output the GS algorithm for sampling purposes as well. We can use the GS algorithm for sampling as follows. If in Algorithm 4.4.1 we substitute (4.8) with

$$\mathbf{Y}_j \sim \kappa_t(\mathbf{y} \,|\, \mathbf{Y}_{j-1}), \quad \mathbf{Y}_0 = \mathbf{X}^t, \quad j = 1, \ldots, s_t,$$

then the resulting sampler is conceptually the same as the GS algorithm (see (3.2)). We call the corresponding sampling algorithm the GS-sampler. Note, however, that in this case $\mathbf{X}^{t+1}$ in Step 3 no longer has pdf (4.9) and $\widehat{c}_{t+1}(\mathbf{X}^t)$ is such that:

$$\mathbb{E}[\widehat{c}_{t+1}(\mathbf{X}^t) \,|\, \mathbf{X}^t] = c_{t+1}(\mathbf{X}^t) := \int I\{S(\mathbf{x}) \geqslant \gamma_{t+1}\} \, \frac{1}{s_t} \sum_{j=1}^{s_t} \kappa_t^j(\mathbf{x} \,|\, \mathbf{X}^t) \, d\mathbf{x},$$

where

$$\kappa_t^j(\mathbf{x} \,|\, \mathbf{y}) := \int \kappa_t(\mathbf{z}_1 \,|\, \mathbf{y}) \kappa_t(\mathbf{z}_2 \,|\, \mathbf{z}_1) \cdots \kappa_t(\mathbf{x} \,|\, \mathbf{z}_{j-1}) \, d\mathbf{z}_1 \cdots d\mathbf{z}_j.$$

As a result of these differences, while convergence of the GS-sampler implies that the sum $\sum_{t=1}^{T-1} \ln(c_{t+1}(\mathbf{X}^t))$ is independent of $\vec{\mathbf{X}}$ (i.e., there are no row effects in matrix C), the converse is not necessarily true. Thus, while the χ^2 diagnostic tests if a sufficient condition for the convergence of the FE-sampler holds, the same diagnostic tests if only a necessary condition for the convergence of the GS-sampler holds.

Example 4.4.1 (Comparison with standard MCMC) To illustrate the performance of the GS-sampler, we consider the problem of sampling from the pdf in Example 4.1.1. We ran Algorithm 4.1.1 with exactly the same setup as in Example 4.1.1, except that the three steps in Algorithm 4.1.2 are executed in a random order, resulting in random Gibbs sampling, as opposed to systematic Gibbs sampling [68]. The random Gibbs sampling ensures that the transition density $\kappa_t(\mathbf{X}^* \,|\, \mathbf{X})$ is reversible [66]. Figure 4.3 shows the empirical distribution of $\mathbf{Z}$ at levels $(\gamma_0, \gamma_1, \gamma_3, \gamma_6) = (-\infty, -117.91, -44.78, 0)$. The $\gamma_0 = -\infty$ case shows the sample from the proposal $p(\mathbf{z})$, and the γ_6 case shows 2030 points approximately distributed from the target density (see Figure 4.1.1). Notice how the two distant modes emerge gradually. The p-value of the χ^2 diagnostic is 0.1, thus failing to detect transient behavior and supporting the hypothesis

that the chain samples according to the target. In addition, the proportion of points in each mode at the final stage is roughly equal to a half, namely, 1009 points belong to the upper right mode and 1021 points belong to the lower left mode. Note that the standard Gibbs sampler applied to $h(\mathbf{z}) = \exp\left(-(z_1^2 + z_2^2 + (z_1 z_2)^2 - 2\lambda z_1 z_2)/2\right)$ fails. In particular, starting from $(0, 0)$ we iterate the following steps 10^9 times.

- Given (Z_1, Z_2), generate $Z_1^* \sim \mathsf{N}\left(\frac{\lambda Z_2}{1+Z_2^2}; \frac{1}{Z_2^2+1}\right)$.
- Given (Z_1^*, Z_2), generate $Z_2^* \sim \mathsf{N}\left(\frac{\lambda Z_1^*}{1+(Z_1^*)^2}; \frac{1}{(Z_1^*)^2+1}\right)$.
- Update $(Z_1, Z_2) := (Z_1^*, Z_2^*)$.

Figure 4.4 shows that the standard Gibbs sampler results in a chain which is trapped in one of the two modes and fails to mix satisfactorily in 10^9 steps.

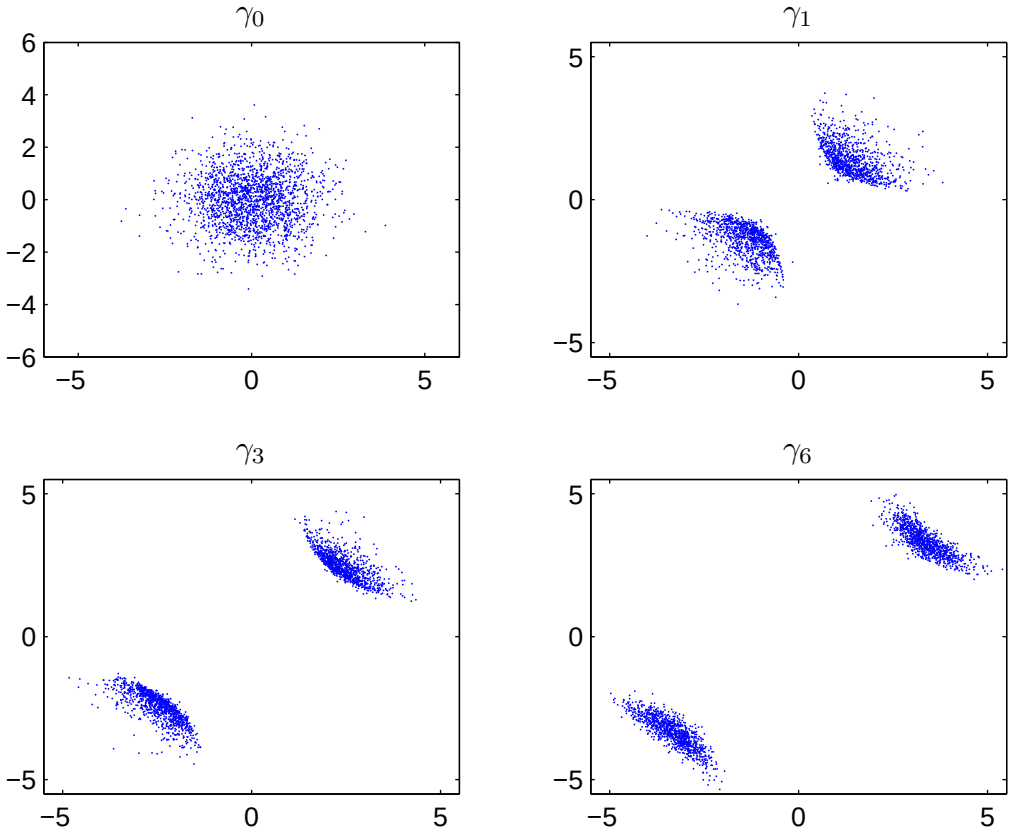

Figure 4.3: The empirical distribution of $\mathbf{Z}$ conditional on $S(\mathbf{X}) \geqslant \gamma_t$ for $t = 0, 1, 3, 6$, respectively.

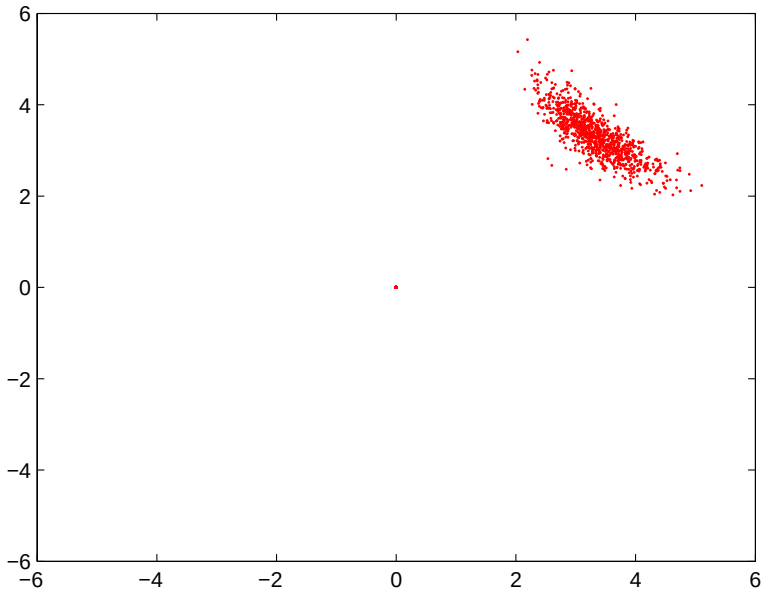

Figure 4.4: Empirical distribution of the output of the standard Gibbs sampler. Here the chain of length 10^9 is thinned to have 10^3 points.

We point out the following differences between the standard Gibbs sampler and the GS-sampler.

1. In comparison with the GS-sampler the Gibbs sampler is simpler to implement on a computer.

2. The performance of the Gibbs sampler is affected by the starting value of the chain. In contrast, the problem of selecting starting values for the chains within the GS-sampler does not exist.

3. The convergence of the Gibbs sampler is difficult to quantify. A consequence of this is that it is usually not know how large the burn-in period has to be. In contrast, the GS-sampler provides a simple χ^2 diagnostic test which makes its performance statistically quantifiable.

4. Our numerical experience suggests that the GS-sampler explores the support of the target density much better than the standard Gibbs sampler.

5. Finally, the GS-sampler provides an unbiased estimate for any quantity of interest. In addition, the variance of the estimate can be computed.

Example 4.4.2 (Comparison with Equi-Energy Sampler) In this example we compare the GS-sampler with the Equi-Energy (EE) sampler [47]. We consider the problem of sampling from the two-dimensional mixture model [47, 53]

$$p(\mathbf{z})H(\mathbf{z}) = \sum_{i=1}^{20} \frac{w_i}{2\pi\sigma_i^2} \exp\left\{-\frac{1}{2\sigma_i^2}(\mathbf{z}-\boldsymbol{\mu}_i)'(\mathbf{z}-\boldsymbol{\mu}_i)\right\},$$

where $\sigma_i = 0.1$, $w_i = 0.05$ for all i, and

$$(\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_{20}) = \begin{pmatrix} 2.18 & 8.67 & 4.24 & 8.41 & 3.93 & 3.25 & 1.7 & 4.59 & 6.91 & 6.87 \\ 5.76 & 9.59 & 8.48 & 1.68 & 8.82 & 3.47 & 0.5 & 5.6 & 5.81 & 5.4 \\ 5.41 & 2.7 & 4.98 & 1.14 & 8.33 & 4.93 & 1.83 & 2.26 & 5.54 & 1.69 \\ 2.65 & 7.88 & 3.7 & 2.39 & 9.50 & 1.50 & 0.09 & 0.31 & 6.86 & 8.11 \end{pmatrix}.$$

Most modes are widely separated by regions of extremely small density (more than 15 standard deviations), making it difficult for the standard Metropolis and Gibbs sampling algorithms to jump across the modes in a reasonable number steps. We now show that the GS-sampler overcomes these problems to produce a population of chains that collectively explore the support of the target pdf fast. We apply Algorithm 4.1.1 with $p(\mathbf{z})H(\mathbf{z})$ given as above and A being the GS algorithm. Note that for this test example the normalizing constant is known in advance ($\mathcal{Z} = 1$). The setup for Algorithm 4.1.1 is as follows. We use the GS algorithm with $N = 10^4$, where the levels and splitting factors are computed using ADAM with $\varrho = 0.1$ and $N = 10^3$. We select a Gaussian proposal $\widetilde{p}(\mathbf{z}) = \frac{1}{2\pi\sigma^2}\mathrm{e}^{-\frac{\mathbf{z}'\mathbf{z}}{2\sigma^2}}$, where the scale σ and the parameters γ, a, b are such that (4.2) holds. To determine a possible tuple (σ, γ, a, b) note that for $\sigma_i \leqslant 1, \forall i$, we have:

$$\begin{aligned} \frac{p(\mathbf{z})H(\mathbf{z})}{\widetilde{p}(\mathbf{z})\mathrm{e}^{a\gamma+b}} &= \sum_{i=1}^{20} \frac{w_i\sigma^2}{\sigma_i^2}\mathrm{e}^{-\frac{(\mathbf{z}-\boldsymbol{\mu}_i)'(\mathbf{z}-\boldsymbol{\mu}_i)}{2\sigma_i^2}+\frac{\mathbf{z}'\mathbf{z}}{2\sigma^2}-a\gamma-b} \\ &\leqslant \frac{\sigma^2}{\min_i\{\sigma_i^2\}}\mathrm{e}^{-(z_1-10)^2-(z_2-10)^2+\frac{\mathbf{z}'\mathbf{z}}{2\sigma^2}-a\gamma-b} \end{aligned}$$

Hence, if we choose $a = b = 1$ and $\mathrm{e}^\gamma = \frac{\sigma^2}{\min_i\{\sigma_i^2\}} = 100\sigma^2$, then $\frac{p(\mathbf{z})H(\mathbf{z})}{\widetilde{p}(\mathbf{z})\mathrm{e}^{a\gamma+b}} \leqslant \mathrm{e}^{-(z_1-10)^2-(z_2-10)^2+\frac{\mathbf{z}'\mathbf{z}}{2\sigma^2}-1}$, which is less than or equal to 1 for all $\mathbf{z}$ provided that $\sigma^2 = \max_{\mathbf{z}} \frac{(z_1^2+z_2^2)/2}{(z_1-10)^2+(z_2-10)^2+1} = 10^2$. Therefore, the tuple $(\sigma, \gamma, a, b) = (10, 4\ln(10), 1, 1)$ ensures that (4.2) holds. With this setup (4.3) can be written as

$$\ell(\gamma) = \mathbb{P}_f\left(\ln(p(\mathbf{Z})H(\mathbf{Z})) - \ln(U) + \frac{\mathbf{Z}'\mathbf{Z}}{2\sigma^2} + \ln(2\pi\sigma^2) - 1 \geqslant \gamma\right),$$

where $f(\mathbf{x}) = \frac{1}{2\pi\sigma^2}\mathrm{e}^{-\frac{\mathbf{z}'\mathbf{z}}{2\sigma^2}} \times I\{0 \leqslant u \leqslant 1\}$. It remains to specify the transition pdf κ_t with stationary density

$$f_t(\mathbf{x}) = \frac{1}{2\pi\sigma^2}\mathrm{e}^{-\frac{\mathbf{z}'\mathbf{z}}{2\sigma^2}} \frac{I\{\ln(p(\mathbf{z})H(\mathbf{z})) - \ln(u) + \frac{\mathbf{z}'\mathbf{z}}{2\sigma^2} + \ln(2\pi\sigma^2) - 1 \geqslant \gamma_t\}}{\ell(\gamma_t)}. \tag{4.12}$$

The transition pdf κ_t consists of running the following Metropolis within Gibbs sampler [66].

Algorithm 4.4.2 (Metropolis within random Gibbs) Given the current state $\mathbf{X} = [\mathbf{Z}, U]'$, execute the steps below in a random order.

- Given $\mathbf{Z}$, draw $U \sim f_t(u \,|\, \mathbf{Z})$.
- Given U, let $\mathbf{Z}$ be the result of 10 steps of the random walk Metropolis Hastings (MH) sampler [68] with target density (4.12) and symmetric proposal $\frac{1}{2\pi\times 5^2}\mathrm{e}^{-\frac{\mathbf{z}'\mathbf{z}}{2\times 5^2}}$.

Output the updated state $\mathbf{X}$.

The result of the above setup is displayed on Figure 4.4, which shows a kernel density estimate of the population of $\mathbf{Z}$ points at the final stage. Note that the levels computed by ADAM are $(\gamma_1, \ldots, \gamma_T) = (-165.71, -2.31, 5.61, 7.92, 4\ln(10))$. The total simulation cost is equivalent to 5.6×10^6 Metropolis-Hastings moves, which is about the same as the cost of running the EE sampler [47], namely, 6×10^6. The χ^2 diagnostic test gives a p-value of 0.85, supporting the hypothesis that population at the final stage is drawn from the target density $p(\mathbf{z})H(\mathbf{z})$. In addition, we conduct a multinomial goodness-of-fit test for the proportion of points that belong to each of the 20 modes. We expect about 1/20 of the points to belong to a particular mode. Each point is assigned to the nearest mode, that is, we use a maximum likelihood estimate. The goodness-of-fit test gave a p-value of 0.16, again supporting the hypothesis the sampler is working as expected. As a further test, we estimate the moments $\{\mathbb{E}Z_i^k,\ i, k = 1, 2\}$ using the population generated at the final stage. The results from the GS algorithm and the EE sampler [47] are displayed in Table 4.7 (upper half). The values in the brackets are the estimated standard deviations. We can conclude that the estimates obtained via the GS-sampler are comparable to the estimates obtained via the EE sampler. Note that as a byproduct of running the GS-sampler we obtain an unbiased estimate of the normalizing constant of the target pdf, namely, $\widehat{\mathcal{Z}}(\gamma) = \widehat{\ell}(\gamma)\ \mathrm{e}^{4\ln(10)+1} = 1.00$, with estimated RE of 3%.

Table 4.7: Comparison between the EE sampler and the GS-sampler. The upper and lower halves correspond to the cases considered in Example 4.4.2 and 4.4.3, respectively. The values in the brackets are the corresponding estimated standard deviations.

	$\mathbb{E}Z_1$	$\mathbb{E}Z_2$	$\mathbb{E}Z_1^2$	$\mathbb{E}Z_2^2$
True value	4.478	4.905	25.605	33.920
EE	4.50 (0.107)	4.94 (0.139)	25.92 (1.098)	34.47 (1.373)
GS	4.48 (0.104)	4.93 (0.155)	25.64 (0.993)	34.37 (1.484)
True value	4.688	5.030	25.558	31.378
EE	4.69 (0.072)	5.03 (0.086)	25.69 (0.739)	31.43 (0.839)
GS	4.68 (0.079)	4.98 (0.099)	25.55 (0.806)	30.87 (0.920)

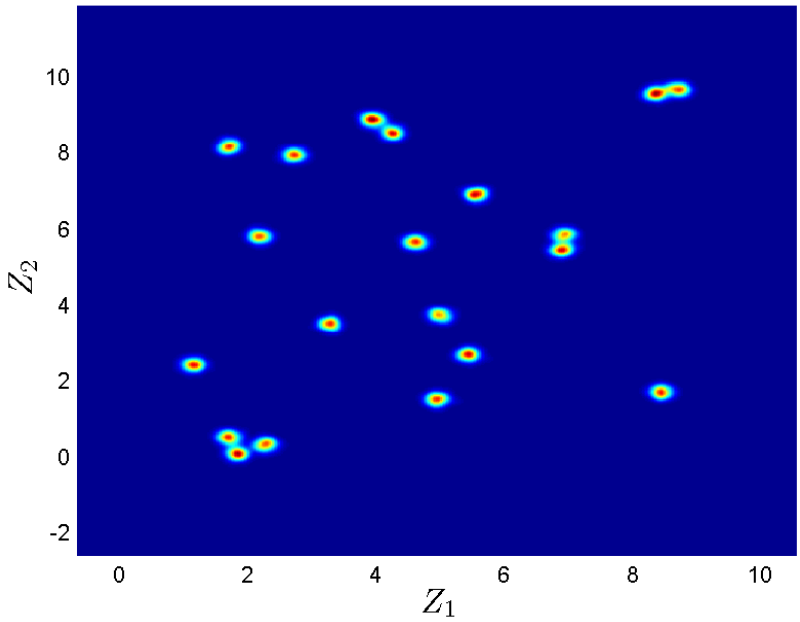

Figure 4.5: Kernel density estimate of the final population (12800 points).

Remark 4.4.4 In cases where it is impossible to deduce analytically a loose upper bound of $\frac{p(\mathbf{z})H(\mathbf{z})}{\widetilde{p}(\mathbf{z})}$ for a given a, b and proposal $\widetilde{p}(\mathbf{z})$, we can use Algorithm 4.1.1 to minimize $-S(\mathbf{x})$ (see Section 4.2), and thus obtain an estimate of the maximum of $\frac{p(\mathbf{z})H(\mathbf{z})}{\widetilde{p}(\mathbf{z})}$. In such a case, there

is no guarantee that (4.2) holds for all $\mathbf{z}$, and as a consequence any estimates obtained using Algorithm 4.1.1 may be biased.

Example 4.4.3 (Example 4.4.2 continued) Consider again Example 4.4.2, but this time with $\sigma_i = d_i/20$ and $w_i \propto 1/d_i$, where $d_i = ||\boldsymbol{\mu}_i - (5,5)'||$. We use the same setup as in Example 4.4.2. Here $\mathrm{e}^{\gamma} = \frac{\sigma^2}{\min_i\{\sigma_i^2\}} \approx \mathrm{e}^{11.24}$ and the ADAM algorithm gives the intermediate levels $(\gamma_1, \ldots, \gamma_T) = (-35.66, 2.08, 5.50, 7.88, 10.36, 11.24)$. The χ^2 diagnostic test gives a p-value of 0.1, supporting the hypothesis that population at the final stage is drawn from the target density $p(\mathbf{z})H(\mathbf{z})$. The estimates in Table 4.7 (lower half) are computed using the population generated at the final stage. The multinomial test goodness-of-fit test gave a p-value of 0.06. The computational cost of our method is the same as in Example 4.4.2. The exact computational cost of the EE sampler could not be determined from the description in [47].

We note the following differences between the GS-sampler and the EE sampler. If the objective of the simulation is the construction of a single chain that easily moves across the whole sample space and jumps across distant modes, then the EE sampler remains the best existing method. The reason for this is that while the EE sampler constructs a single rapidly mixing chain (with the help of an interacting population of chains), the GS-sampler runs a population of chains, started in a different location of the sample space. Thus, in the GS-sampler a population of chains explores the sample space, instead of a single rapidly mixing chain. If, however, the objective of the simulation is parameter estimation or the generation of an approximate sample according to the target density, then the GS-sampler has several advantages.

First, the GS-sampler provides a p-value to quantify the performance of the sampler. In contrast, there is no easy way to test the convergence of the EE sampler apart from a subjective graphical assessment of the output. For example, we may monitor whether all the modes of the target density are visited often enough, but there is no p-value to quantify the meaning of "often enough". As a result it is impossible to know if the burn-in period of the EE chain is long enough.

Second, the splitting approach provides unbiased estimates of many quantities of interest (e.g., the partition function $\mathcal{Z}(\gamma)$), regardless of the sample size or the convergence of the underlying Markov chains. In contrast, the EE sampler can only provide ergodic estimates that rely on the asymptotic convergence of the underlying chain.

Third, while the EE sampler requires the selection of initializing states, a problem typical for all MCMC algorithms, the GS-sampler does not require the user to specify any initializing states.

Finally, while the GS-sampler requires the tuning of parameters N (sample size) and ϱ (rarity parameter), and the MH proposal in Algorithm 4.4.2, the EE sampler requires the tuning of a much larger number of parameters. For instance, Examples 4.4.2 and 4.4.3 require the tunning of the following parameters [47].

1. The parameter K, which is the number of energy and temperature levels.
2. Given the lowest and highest energy levels E_0 and E_K, the user has to specify the intermediate energy levels $E_1, \ldots, E_{K-1}$ and the temperature levels $\{T_i\}_{i=0}^K$.
3. Associated with every temperature level T_i in the EE sampler is a MH proposal. For all proposals the MH acceptance ratio has to be in the range $(0.22, 0.32)$. Although this tuning can in principle be automated, a practical implementation requires specifying a mechanism for the dynamic monitoring and adjustment of the MH proposals so that the MH acceptance ratio (over a user-specified time window) lies approximately in the user-prescribed range.
4. The equi-energy jump probability p_{ee}, which controls the jumping across the modes of the target density, has to be tuned. For example, when $p_{\text{ee}} > 0.4$ the performance of the EE sampler worsens in the examples above [47].

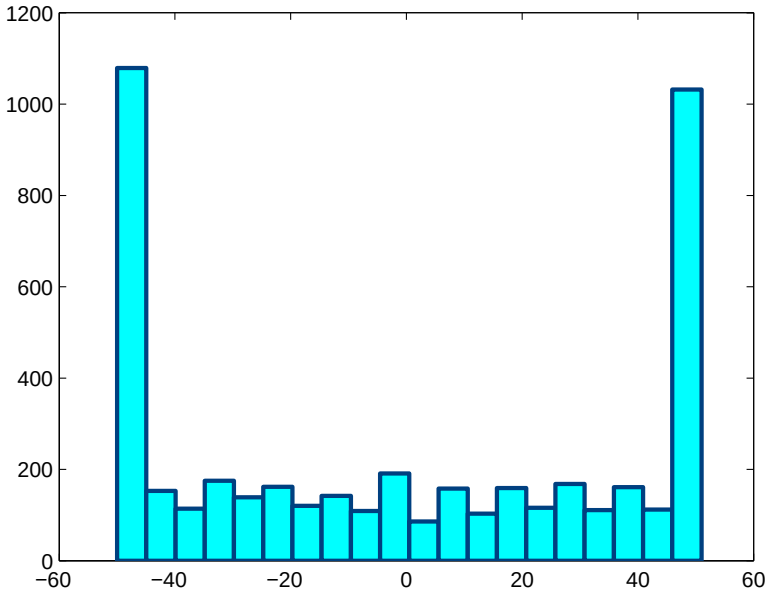

Figure 4.6: A histogram constructed from the empirical distribution of the total magnetization $M = \sum_{n=1}^{50} Z_n$.

Example 4.4.4 (Sampling over a countable space) Consider the problem of sampling from the Boltzmann pdf $h(\mathbf{z})/\mathcal{Z}(\lambda)$ in Example 4.3.1 for $\lambda = 2$. We use exactly the same setup as in Example 4.3.1, except that the components of the vector $\mathbf{X} = [\mathbf{Z}, U]$ are updated in a random order within Algorithm 4.3.2. The randomized updating ensures that the transition density $\kappa_t(\mathbf{x}^* \,|\, \mathbf{x})$ is reversible [66]. The p-value of the χ^2 diagnostic is 0.17, supporting the hypothesis that the population is drawn from the target $h(\mathbf{z})/\mathcal{Z}(2)$. In addition, we assess the convergence graphically. Figure 4.6 shows the histogram of the empirical distribution of the *total magnetization* $M = \sum_{n=1}^{50} Z_n$ of the Ising model. Liu [54] shows that exact simulation from the target pdf yields a histogram with the same features as the ones depicted on Figure 4.6. Based on such graphical evidence Liu diagnoses convergence.

4.5 Improved Importance Sampling

In this section we show that significant variance reduction can be achieved when a splitting algorithm is used to determine the parameters of a parametric Importance Sampling (IS) density, which is then used to estimate (4.1) via an IS estimator. The idea is as follows [6]. Let $g(\mathbf{z}; \boldsymbol{\lambda})$ be an IS density parametrized by $\boldsymbol{\lambda} \in \Lambda$, and such that $g(\mathbf{z}; \boldsymbol{\lambda}) = 0 \;\Rightarrow p(\mathbf{z})H(\mathbf{z}) = 0$. Without loss of generality we assume $H(\mathbf{z}) \geqslant 0$. Then, we have the unbiased IS estimator of (4.1):

$$\widehat{\mathcal{Z}}_1 = \frac{1}{N_1} \sum_{i=1}^{N_1} \frac{p(\mathbf{Z}_i)H(\mathbf{Z}_i)}{g(\mathbf{Z}_i; \boldsymbol{\lambda})}, \quad \mathbf{Z}_1, \ldots, \mathbf{Z}_{N_1} \sim_{\text{iid}} g(\mathbf{z}; \boldsymbol{\lambda}). \tag{4.13}$$

The performance of the IS estimator depends crucially on the choice of the parameter $\boldsymbol{\lambda}$ [67, 68]. A common approach to determining an optimal value for $\boldsymbol{\lambda}$ is to select $\boldsymbol{\lambda}$ so that the IS density g is as close as possible to the *minimum-variance* IS density [68] for the estimation of (4.1), namely, $g^*(\mathbf{z}) = \frac{p(\mathbf{z})H(\mathbf{z})}{\mathcal{Z}}$. The "closeness" between g and g^* is measured by a suitable ϕ-divergence distance measure. Specifically, the optimal $\boldsymbol{\lambda}$ is given by:

$$\boldsymbol{\lambda}^* = \underset{\boldsymbol{\lambda} \in \Lambda}{\operatorname{argmin}} \int g^*(\mathbf{z}) \, \phi\left(\frac{g(\mathbf{z}; \boldsymbol{\lambda})}{g^*(\mathbf{z})}\right) d\mathbf{z},$$

where $\phi : \mathbb{R}_+ \to \mathbb{R}$ is twice continuously differentiable, and $\phi(1) = 0$, $\phi''(z) > 0, \forall z > 0$. Choosing $\phi(z) = \ln(z)$ gives the Cross Entropy (CE) method [67] for the optimal selection of $\boldsymbol{\lambda}$, and choosing $\phi(z) = 1/z$ gives the Variance Minimization method. Here we consider the CE method, in which the minimization above simplifies to:

$$\boldsymbol{\lambda}^* = \underset{\boldsymbol{\lambda} \in \Lambda}{\operatorname{argmax}} \int p(\mathbf{z})H(\mathbf{z}) \ln(g(\mathbf{z}; \boldsymbol{\lambda})) \, d\mathbf{z}.$$

In practice, $\mathcal{Z}(\boldsymbol{\lambda}) := \int p(\mathbf{z}) H(\mathbf{z}) \ln(g(\mathbf{z};\boldsymbol{\lambda}))\, d\mathbf{z}$ has to be estimated using adaptive IS techniques [67], which are not always successful. Here we take a different approach and use Algorithm 4.3.1 with $W(\mathbf{z};\boldsymbol{\lambda}) = \ln(g(\mathbf{z};\boldsymbol{\lambda}))$ to estimate the function $\mathcal{Z}(\boldsymbol{\lambda})$. In effect, we treat $\boldsymbol{\lambda}$ as a sensitivity parameter. Then, we use

$$\widehat{\boldsymbol{\lambda}^*} = \operatorname*{argmax}_{\boldsymbol{\lambda}\in\Lambda} \widehat{\mathcal{Z}}(\boldsymbol{\lambda})$$

in (4.13) to give the final IS estimator of $\mathcal{Z}$. Proposition 4.1.1 ensures that the estimate $\widehat{\mathcal{Z}}(\boldsymbol{\lambda})$ is unbiased and hence $\widehat{\boldsymbol{\lambda}^*}$ is a reasonable approximation to the true $\boldsymbol{\lambda}^*$.

Example 4.5.1 (Example 3.1.1 continued) We apply the procedure described above to the SAT counting problem as follows. We estimate ℓ defined in (3.6) via the IS estimator ($f(\mathbf{x}) = 1/2^n,\ \mathbf{x} \in \{0,1\}^n$):

$$\widehat{\ell}_1 = \frac{1}{N_1} \sum_{i=1}^{N_1} \frac{f(\mathbf{X}_i) I\{S(\mathbf{X}_i) \geqslant m\}}{g(\mathbf{X}_i;\boldsymbol{\lambda})}, \quad \mathbf{X}_1,\ldots,\mathbf{X}_{N_1} \sim_{\text{iid}}\ g(\mathbf{x};\boldsymbol{\lambda}), \tag{4.14}$$

where the IS density $g(\,\cdot\,;\boldsymbol{\lambda})$ is the multivariate Bernoulli mixture:

$$g(\mathbf{x};\boldsymbol{\lambda}) = \sum_{k=1}^{K} w_k \prod_{j=1}^{n} p_{kj}^{x_j}\ (1-p_{kj})^{1-x_j}.$$

Here K is the number of mixture components, $\mathbf{w} = (w_1,\ldots,w_K)^T,\ \sum_{k=1}^K w_k = 1,\ w_k \geqslant 0$, are the weights associated with each component, each $\mathbf{p}_k = (p_{k1},\ldots,p_{kn})$ is a vector of probabilities, and $\boldsymbol{\lambda} = (\mathbf{w},\mathbf{p}_1,\ldots,\mathbf{p}_K)$ collects all the unknown parameters. We assume that K is known. Then, the optimal $\boldsymbol{\lambda}$ is given by

$$\boldsymbol{\lambda}^* = \operatorname*{argmax}_{\boldsymbol{\lambda}} \sum_{\mathbf{x}\in\{0,1\}^n} f(\mathbf{x}) I\{S(\mathbf{x}) \geqslant m\} \ln(g(\mathbf{x};\boldsymbol{\lambda})).$$

Therefore, we use the estimate

$$\widehat{\boldsymbol{\lambda}^*} = \operatorname*{argmax}_{\boldsymbol{\lambda}} \widehat{\ell}_{\boldsymbol{\lambda}}(m), \tag{4.15}$$

where $\widehat{\ell}_{\boldsymbol{\lambda}}(m)$ is provided by Algorithm 4.3.1 with $\gamma = m$ and $W_{\boldsymbol{\lambda}}(\mathbf{x}) = \ln(g(\mathbf{x};\boldsymbol{\lambda}))$. Program (4.15) is identical to maximizing the likelihood function of the Bernoulli mixture model based on the final population of the GS-sampler. Despite the fact that there is no closed form analytical solution to the program (4.15), as it typical for the CE method [67], we use a multivariate Bernoulli mixture since it is one of the most flexible parametric models on a binary space. In addition, likelihood optimization routines such as the EM algorithm or K-means clustering

[61, 62] are quite effective in solving program (4.15). Thus, for a given K, we approximate the global maximizer in (4.15) using the EM algorithm. For all computed solutions, if a given estimate $\widehat{p}_{kj}$ happens to be zero or one, we automatically set it to 0.001 or 0.999, respectively. This ensures that $g(\mathbf{z};\boldsymbol{\lambda}) = 0 \;\Rightarrow p(\mathbf{z})H(\mathbf{z}) = 0$ for all $\mathbf{z}$.

As a numerical example consider Table 4.8 which shows the simulation results for a number of SAT counting problems. The table displays K and N_1, as used to compute the IS estimator (4.14), and N — the sample size used in Algorithm 4.3.1 for the computation of (4.15). In all cases we used $\varrho = 0.5$ to determine appropriate weights and levels for Algorithm 4.3.1. Note the small relative error of the estimates in column three. We are not aware of any existing algorithms that provide estimates of similar accuracy for the SAT counting problem.

Table 4.8: 12 SAT counting problems via IS estimator (4.14). In all cases $\varrho = 0.5$.

Instance	$\widehat{\lvert\mathscr{X}^*\rvert}$	RE	N_1	N	K
uf75-01	2258.28	0.03%	10^8	10^4	40
uf75-02	4590.02	0.07%	10^8	10^4	40
uf250-01	3.38×10^{11}	4.4%	10^8	10^4	40
RTI_k3_n100_m429_0	20943.79	0.01%	10^8	10^4	40
RTI_k3_n100_m429_1	24541.70	0.02%	10^8	10^4	40
RTI_k3_n100_m429_2	3.9989	0.01%	10^8	10^3	10
RTI_k3_n100_m429_3	376.016	0.01%	10^8	10^3	16
RTI_k3_n100_m429_4	4286.28	0.3%	10^7	10^3	10
RTI_k3_n100_m429_5	7621.11	0.7%	10^7	10^3	30
RTI_k3_n100_m429_6	2210.20	0.01%	10^8	10^4	40
RTI_k3_n100_m429_7	1869.64	0.3%	10^8	10^4	40
RTI_k3_n100_m429_8	1832.29	0.01%	10^8	10^4	40

We point out that the parameters K were determined experimentally. Naturally, the larger the value of K, the more accurate the IS estimate. A larger value for K, however, incurs greater computational cost in solving program (4.15) and evaluating the IS estimator (4.13). We found that values of K as low as 10 result in estimates with satisfactory RE (less than 5%). Note that the optimal selection of the number of components in a mixture model is still a contentious problem without a satisfactory solution [62].

Chapter 5

Splitting Methods Synopsis

In this chapter we draw conclusions and point to possible future work regarding the applications and extensions of the GS algorithm.

This thesis presents a Generalized Splitting algorithm, that extends the original splitting idea of Kahn and Harris to static and non-Markovian simulation problems. Similar to the original splitting method, the GS method induces a branching process by constructing an artificial Markov chain via the Gibbs or Metropolis-Hastings samplers. Notable features of the proposed approach are as follows.

First, the GS algorithm provides an unbiased and consistent estimator of ℓ in (1.1) without requiring that the Markov chain constructed by the GS algorithm reaches stationarity or mixes well. It is not even necessary that the chain is irreducible. In contrast, standard MCMC algorithms provide a biased estimate of ℓ for any finite run time. In general, this bias can only be reduced by discarding observations during the initial transient or burn-in phase of the MCMC algorithms. Thus, any inference is always based upon a portion of the sampler output. In addition, the GS algorithm provides an unbiased estimate of the mean square error of $\widehat{\ell}$. In contrast, standard MCMC algorithms provide biased error estimates. The lack of unbiasedness forces the practitioner to rely on the large sample ergodic properties of the Markov chain to both eliminate the unknown bias of the ergodic estimator and provide error estimates. In practice, one can never be sure if the sample is large enough to eliminate the unknown bias.

Second, the GS algorithm can be used to generate samples (approximately) according to a given multidimensional pdf, for which standard MCMC methods fail by significantly improving the exploration of the multidimensional pdf. In addition, the GS-sampler provides a computationally inexpensive convergence diagnostic based on a χ^2 test statistic. In contrast, one of the most difficult problems in the implementation of standard MCMC methods is quantifying when stationarity has been achieved. Most of the existing convergence diagnostics [9] are computationally intensive and graphical in nature.

Third, there are no problems associated with selecting appropriate starting values for the Markov chains within the GS algorithm. In contrast, the performance of MCMC algorithms can be affected by the starting values when the mixing of the chain is not fast enough.

Fourth, combining the GS algorithm with standard importance sampling, we obtain estimates for the SAT counting problem with unprecedented low relative error.

Finally, as pointed out in Section 4.4, Proposition 4.4.1 corrects the statement in our paper, [7], that the ADAM algorithm can be used for exact (as opposed to approximate) Markov chain sampling.

Possible future extensions of the proposed methodology are directly suggested by the existing results on the original splitting method.

First, since the original splitting method can be implemented in a memory efficient way using the so-called Global Step approach, the same memory efficient implementation may benefit the GS algorithm.

Second, the concept of the so-called importance function plays a central role in the classical splitting method. The extension of the idea to the generalized case leads one to consider estimating the conditional probabilities $\mathbb{P}(h_{t+1}(\mathbf{X}) \geqslant \gamma \,|\, h_t(\mathbf{X}) \geqslant \gamma)$ for some sequence of importance functions $\{h_t\}_{t=0}^T$, such that $h_T(\mathbf{X}) = S(\mathbf{X})$ and the events $\{h_t(\mathbf{X}) \geqslant \gamma\} \subseteq \{h_{t+1}(\mathbf{X}) \geqslant \gamma\}$ for all t. In this thesis we have only considered the special case where $h_t(\mathbf{X}) = S(\mathbf{X})/a_t$ (that is, $\mathbb{P}(S(\mathbf{X}) \geqslant \gamma_{t+1} \,|\, S(\mathbf{X}) \geqslant \gamma_t)$ with $\gamma_t = a_t\gamma$) for some suitably chosen sequence $\{a_t\}$. It is possible that by selecting a different importance function one can achieve even better efficiency gains.

Third, since the use of quasi-Monte Carlo methods in combination with the classical splitting method has been shown to be quite effective, one may consider extending the use of quasi-Monte Carlo methods to the setting the GS algorithm.

Finally, we note that the splitting method for rare-event simulation can be viewed as a particular Sequential Importance Sampler in a higher dimensional space constructed using auxiliary variables [64]. However, the formulation of the splitting method as a type of sequential importance sampling method can be unnecessarily abstract and does not give a clearly defined practicable algorithm [64] in our setting. In addition, the sequential importance sampling framework [64] does not make it clear how to obtain unbiased estimates of ℓ. We believe that the splitting methods described here can be improved by examining them within the sequential importance sampling framework and using many of the results already known in the sequential importance sampling literature.

Part II

Kernel Density Estimation via Diffusion

Chapter 6

Introduction to part Two

In this chapter we make a brief overview of kernel density estimators and point to some of their drawbacks such as unreliable bandwidth selection algorithms and boundary bias.

Nonparametric density estimation is an important tool in the statistical analysis of data. A nonparametric estimate can be used, for example, to assess the multimodality, skewness, or any other structure in the distribution of the data [71, 73]. It can also be used for the summarization of Bayesian posteriors, classification and discriminant analysis [74]. Nonparametric density estimation has even proved useful in Monte Carlo computational methods, such as the smoothed bootstrap method and the particle filter method [19]. Nonparametric density estimation is an alternative to the parametric approach of Fisher, in which one specifies a model up to a small number of parameters and then estimates the parameters optimally via the likelihood principle. The advantage of the nonparametric approach is that it offers a far greater flexibility in modeling a given dataset and, unlike the classical approach, is not affected by specification bias [52]. Currently the most popular nonparametric approach to density estimation is *kernel density estimation* (see [71, 74, 79]).

Despite the vast body of literature on the subject, there are still many contentious issues regarding the implementation and practical performance of kernel density estimators. First, the most popular data-driven bandwidth selection technique, the *plug-in* method [39, 72], is adversely affected by the so-called *normal reference rule* [18, 38], which is essentially a construction of a preliminary normal model of the data upon which the performance of the bandwidth selection method depends. Although plug-in estimators perform well when the normality assumption holds approximately, at a conceptual level the use of the normal reference rule invalidates the original motivation for applying a nonparametric method in the first place.

Second, the popular Gaussian kernel density estimator [59] lacks *local adaptivity*, and this often results in a large sensitivity to outliers, the presence of spurious bumps, and in an overall unsatisfactory bias performance — a tendency to flatten the peaks and valleys of the density [75].

Third, most kernel estimators suffer from *boundary bias* when, for example, the data is nonnegative — a phenomenon due to the fact that most kernels do not take into account specific knowledge about the domain of the data [58, 65].

These problems have been alleviated to a certain degree by the introduction of more sophisticated kernels than the simple Gaussian kernel. Higher-order kernels have been used as a way to improve local adaptivity and reduce bias [40], but these have the disadvantages of not giving proper nonnegative density estimates, and of requiring a large sample size for good performance [59]. The lack of local adaptivity has been addressed by the introduction of *adaptive* kernel estimators [1, 30, 31, 60]. These include the *balloon* estimators, *nearest neighbor* estimators and *variable bandwidth* kernel estimators [55, 75], none of which yield bona fide densities, and thus remain somewhat unsatisfactory. Other proposals such as the *sample point adaptive* estimators are computationally burdensome (the fast Fourier transform cannot be applied [73]), and in some cases do not integrate to unity [65]. The *boundary kernel estimators* [37], which are specifically designed to deal with boundary bias, are either not adaptive away from the boundaries or do not result in bona fide densities [36]. Thus the literature abounds with partial solutions that obscure a unified comprehensive framework for the resolution of these problems.

The aim of this second part of the thesis is to introduce an adaptive kernel density estimation method based on the smoothing properties of linear diffusion processes. The key idea is to view the kernel from which the estimator is constructed as the transition density of a diffusion process. We utilize the most general linear diffusion process that has a given limiting and stationary probability density. This stationary density is selected to be either a pilot density estimate or a density that the statistician believes represents the information about the data prior to observing the available empirical data. The approach leads to a simple and intuitive kernel estimator with substantially reduced asymptotic bias and mean square error. The proposed estimator deals well with boundary bias and, unlike other proposals, is always a bona fide probability density function. We show that the proposed approach brings under a single framework some well-known bias reduction methods, such as the Abramson estimator [1] and other variable location or scale estimators [15, 32, 70, 60].

In addition, the thesis introduces an improved plug-in bandwidth selection method that completely avoids the normal reference rules [38] that have aversely affected the performance of plug-in methods. The new plug-in method is thus genuinely "nonparametric", since it does not require a preliminary normal model for the data. Moreover, our plug-in approach does not involve numerical optimization and is not much slower than computing a normal reference rule [4].

The rest of the second part of the thesis is organized as follows. First, in chapter 7 we describe the Gaussian kernel density estimator and explain how it can be viewed as a special

case of smoothing using a diffusion process. The Gaussian kernel density estimator is then used in chapter 8 to motivate the most general linear diffusion that will have a set of essential smoothing properties. We analyze the asymptotic properties of the resulting estimator in chapter 8, section 8.2. In chapter 9 we explain how to compute the asymptotically optimal plug-in bandwidth for both the Gaussian and diffusion kernel density estimators. In section 9.4 the practical benefits of the model are demonstrated through simulation examples on some well-known datasets [59]. Our findings demonstrate an improved bias performance and low computational cost, and a boundary bias improvement. Finally, in chapter 10 we summarize our findings and point to possible direction of future research.

Chapter 7

Density Estimation with a Gaussian kernel

In this chapter we introduce the diffusion kernel density estimator as a generalization of a nonparametric density estimator with Gaussian kernel. We explain how the diffusion kernel tackles the problems of boundary bias.

Given N independent realizations $\mathcal{X}_N \equiv \{X_1, \ldots, X_N\}$ from an unknown continuous probability density function (pdf) f on $\mathscr{X}$, the *Gaussian kernel density estimator* is defined as:

$$\widehat{f}(x;t) = \frac{1}{N}\sum_{i=1}^{N}\phi(x, X_i; t), \quad x \in \mathbb{R}, \tag{7.1}$$

where

$$\phi(x, X_i; t) = \frac{1}{\sqrt{2\pi t}}\, \mathrm{e}^{-\frac{(x-X_i)^2}{2t}}$$

is a Gaussian pdf (kernel) with location X_i and scale $\sqrt{t}$. The scale is usually referred to as the *bandwidth*. Much research has been focused on the optimal choice of t in (7.1), because the performance of $\widehat{f}$ as an estimator of f depends crucially on its value [39, 72]. A well-studied criterion used to determine an optimal t is the *Mean Integrated Squared Error* (MISE):

$$\mathrm{MISE}\{\widehat{f}\}(t) = \mathbb{E}_f \int [\widehat{f}(x;t) - f(x)]^2 dx,$$

which is conveniently decomposed into integrated squared bias and integrated variance components:

$$\mathrm{MISE}\{\widehat{f}\}(t) = \int \Big(\underbrace{\mathbb{E}_f[\widehat{f}(x;t)] - f(x)}_{\text{pointwise bias of } f} \Big)^2 dx + \int \underbrace{\mathrm{Var}_f[\widehat{f}(x;t)]}_{\text{pointwise variance of } f} dx.$$

Note that the expectation and variance operators apply to the random sample $\mathcal{X}_N$. The MISE depends on the bandwidth $\sqrt{t}$ and f in a quite complicated way. The analysis is simplified when one considers the asymptotic approximation to the MISE, denoted AMISE, under the consistency requirements that $t = t_N$ depends on the sample size N such that $t_N \downarrow 0$ and $N\sqrt{t_N} \to \infty$ as $N \to \infty$, and f is twice continuously differentiable [72]. The asymptotically

optimal bandwidth is then the minimizer of the AMISE. The asymptotic properties of (7.1) under these assumptions are summarized in Appendix A.1.

A key observation about the Gaussian kernel density estimator (7.1) is that it is the unique solution to the diffusion partial differential equation (PDE):

$$\frac{\partial}{\partial t}\widehat{f}(x;t) = \frac{1}{2}\frac{\partial^2}{\partial x^2}\widehat{f}(x;t), \quad x \in \mathscr{X},\ t > 0, \tag{7.2}$$

with $\mathscr{X} \equiv \mathbb{R}$ and initial condition $\widehat{f}(x;0) = \Delta(x)$, where $\Delta(x) = \frac{1}{N}\sum_{i=1}^{N}\delta(x - X_i)$ is the empirical density of the data $\mathcal{X}_N$ (here $\delta(x - X_i)$ is the Dirac measure at X_i). Equation (7.2) is the well-known Fourier heat equation [49]. This link between the Gaussian kernel density estimator and the Fourier heat equation has been noted in Chaudhuri and Marron [14]. We will, however, go much further in exploiting this link. In the heat equation interpretation, the Gaussian kernel in (7.1) is the so-called Green's function [49] for the diffusion PDE (7.2). Thus the Gaussian kernel density estimator $\widehat{f}(x;t)$ can be obtained by evolving the solution of the parabolic PDE (7.2) up to time t.

To illustrate the advantage of the PDE formulation over the more traditional formulation (7.1), consider the case where the domain of the data is *known to be* $\mathscr{X} \equiv [0,1]$. It is difficult to see how (7.1) can be easily modified to account for the finite support of the unknown density. Yet, within the PDE framework, all we have to do is solve the diffusion equation (7.2) over the finite domain $[0,1]$ with initial condition $\Delta(x)$ and the Neumann boundary condition

$$\frac{\partial}{\partial x}\widehat{f}(x;t)\Big|_{x=1} = \frac{\partial}{\partial x}\widehat{f}(x;t)\Big|_{x=0} = 0.$$

The boundary condition ensures that $\frac{d}{dt}\int_{\mathscr{X}}\widehat{f}(x;t)\,dx = 0$, from where it follows that $\int_{\mathscr{X}}\widehat{f}(x;t)\,dx = \int_{\mathscr{X}}\widehat{f}(x;0)\,dx = 1$ for all $t \geqslant 0$. The analytical solution of this PDE in this case is [3]:

$$\widehat{f}(x;t) = \frac{1}{N}\sum_{i=1}^{N}\kappa(x, X_i; t), \quad x \in [0,1], \tag{7.3}$$

where the kernel κ is given by

$$\kappa(x, X_i; t) = \sum_{k=-\infty}^{\infty}\phi(x, 2k + X_i; t) + \phi(x, 2k - X_i; t), \quad x \in [0,1]. \tag{7.4}$$

Thus, the kernel accounts for the boundaries in a manner similar to the boundary correction of the *reflection method* [73]. We now compare the properties of the kernel (7.4) with the properties of the Gaussian kernel ϕ in (7.1).

First, the series representation (7.4) is useful for deriving the small bandwidth properties of the estimator in (7.3). The asymptotic behavior of $\kappa(x, X_i; t)$ as $t \to 0$ in the interior of the domain $[0, 1]$ is no different from that of the Gaussian kernel, namely,

$$\sum_{k=-\infty}^{\infty} \phi(x, 2k + X_i; t) + \phi(x, 2k - X_i; t) \sim \phi(x, X_i; t), \quad t \downarrow 0,$$

for any fixed x in the interior of the domain $[0, 1]$. Here $q(t) \sim z(t),\ t \downarrow t_0$ stands for $\lim_{t \downarrow t_0} \frac{q(t)}{z(t)} = 1$. Thus, for small t, the estimator (7.3) behaves like the Gaussian kernel density estimator (7.1) in the interior of $[0, 1]$. Near the boundaries at $x = 0, 1$, however, the estimator (7.3) is consistent, while the Gaussian kernel density estimator is inconsistent. In particular, a general result in Appendix A.4 includes as a special case the following boundary property of the estimator (7.3):

$$\mathbb{E}_f \widehat{f}(x_N; t_N) = f(x_N) + O(\sqrt{t_N}), \quad N \to \infty,$$

where $x_N = \alpha\, t_N$ for some $\alpha \in [0, 1]$, and $t_N \downarrow 0$ as $N \to \infty$. This shows that (7.3) is consistent at the boundary $x = 0$. Similarly, (7.3) can be shown to be consistent at the boundary $x = 1$. In contrast, the Gaussian kernel density estimator (7.1) is inconsistent [79] in the sense that,

$$\mathbb{E}_f \widehat{f}(0; t_N) = \frac{1}{2} f(0) + O(\sqrt{t_N}), \quad N \to \infty.$$

The large bandwidth behavior ($t \to \infty$) of (7.3) is obtained from the following equivalent expression for (7.4) (see [3]):

$$\kappa(x, X_i; t) = \sum_{k=-\infty}^{\infty} \mathrm{e}^{-k^2 \pi^2 t/2} \cos(k\pi x) \cos(k\pi X_i). \tag{7.5}$$

From (7.5) we immediately see that

$$\kappa(x, X_i; t) \sim 1 + 2\, \mathrm{e}^{-\pi^2 t/2} \cos(\pi x) \cos(\pi X_i), \quad t \to \infty,\ x \in [0, 1]. \tag{7.6}$$

In other words, as the bandwidth becomes larger and larger, the kernel (7.4) approaches the uniform density on $[0, 1]$.

Remark 7.0.1 An important property of the estimator (7.3) is that the number of local maxima or modes is a non-increasing function of t. This follows from the *maximum principle* for parabolic PDE, see, e.g., [49].

For example, a necessary condition for a local maximum at, say, $(x_0, t_0),\ t_0 > 0, x_0 \in (0, 1)$ is $\frac{\partial^2}{\partial x^2} \widehat{f}(x_0; t_0) \leqslant 0$. From (7.2), this implies $\frac{\partial}{\partial t} \widehat{f}(x_0; t_0) \leqslant 0$, from which it follows that there

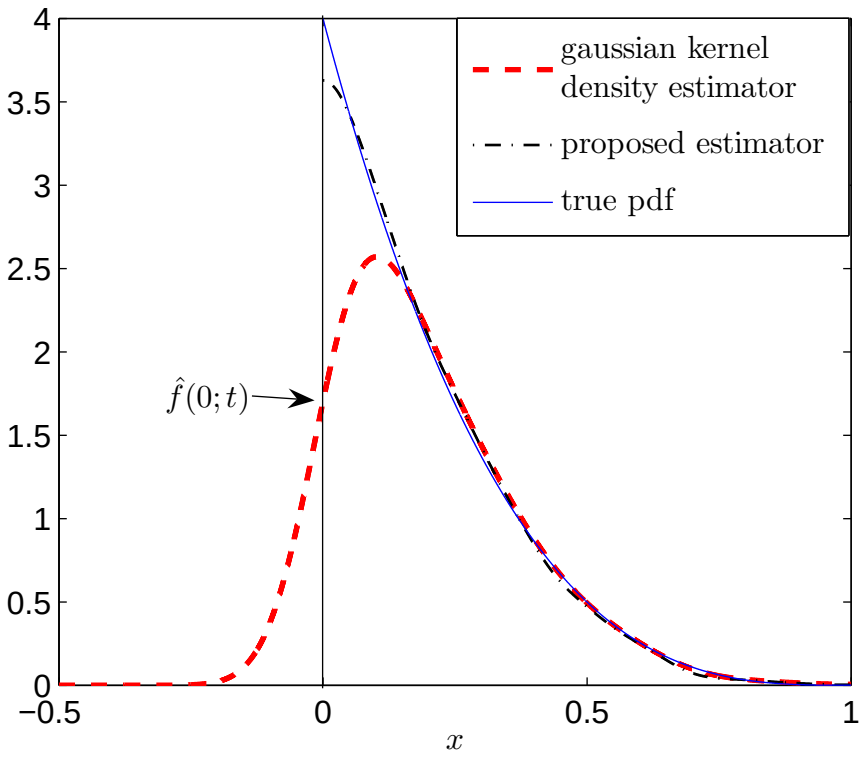

Figure 7.1: Boundary bias in the neighborhood of $x = 0$.

exists an $\varepsilon > 0$ such that $\widehat{f}(x_0; t_0) \geqslant \widehat{f}(x_0; t_0 + \varepsilon)$. As a consequence of this, as t becomes larger and larger, the number of local maxima of (7.3) is a non-increasing function. This property is shared by the Gaussian kernel density estimator (7.1) and has been exploited in various ways by Silverman [73].

Example 7.0.2 Figure 7.1 gives an illustration of the performance of estimators (7.3) and (7.1), where the true pdf is the beta density $4(1-x)^3$, $x \in [0,1]$, and the estimators are build from a sample of size $N = 1000$ with a common bandwidth $\sqrt{t} = 0.05248$. This is the asymptotically optimal (see (A.4) in Appendix A.1) bandwidth for the Gaussian kernel density estimator (7.1). Note that the Gaussian kernel density estimator is close to half the value of the true pdf at the boundary $x = 0$. Overall, the diffusion estimator (7.3) is much closer to the true pdf. The proposed estimator (7.3) appears to be the first kernel estimator that is consistent at all boundaries and at the same time remains a genuine pdf, that is, is nonnegative and integrates to one. Existing boundary correction methods [33, 44, 45] either account for the bias at a single end-point, or the resulting estimators are not genuine pdfs.

Remark 7.0.2 In applications such as the smoothed bootstrap [19], there is a need for efficient

random variable generation from the kernel density estimate. Generation of random variables from the kernel (7.4) is easily accomplished using the following procedure. Generate $Z \sim \mathsf{N}(0, t)$ and let $Y = X_i + Z$. Compute $W = Y \mod 2$, and let $X = |W|$. Then it is easy to show (e.g., using characteristic functions) that X has the density given by (7.4).

Given the nice boundary bias properties of the estimator that arises as the solution of the diffusion PDE (7.2), it is of interest to investigate if equation (7.2) can be somehow modified or generalized to arrive at an even better kernel estimator. This motivates us to consider in the next chapter the most general linear time-homogeneous diffusion PDE as a starting point for the construction of a better kernel density estimator.

Chapter 8

The Diffusion Kernel Density Estimator

In this chapter we introduce the diffusion density estimator and derive its asymptotic bias and variance. In addition, we show how the diffusion estimator subsumes many variable location-scale kernel density estimators as special cases.

8.1 The Diffusion Estimator

Our extension of the simple diffusion model (7.2) is based on the smoothing properties of the linear diffusion PDE

$$\frac{\partial}{\partial t} g(x;t) = Lg(x;t), \quad x \in \mathscr{X},\ t > 0, \tag{8.1}$$

where the linear differential operator L is of the form $\frac{1}{2}\frac{d}{dx}\left(a(x)\frac{d}{dx}\left(\frac{\cdot}{p(x)}\right)\right)$, and a and p can be any arbitrary positive functions on $\mathscr{X}$ with bounded second derivatives, and the initial condition is $g(x,0) = \Delta(x)$. If the set $\mathscr{X}$ is bounded, we add the boundary condition $\frac{\partial}{\partial x}\left(\frac{g(x;t)}{p(x)}\right) = 0$ on $\partial\mathscr{X}$, which ensures that the solution of (8.1) integrates to unity. The PDE (8.1) describes the pdf of X_t for the Itô diffusion process $(X_t, t > 0)$ given by [20]:

$$dX_t = \mu(X_t)dt + \sigma(X_t)dB_t, \tag{8.2}$$

where the drift coefficient $\mu(x) = \frac{a'(x)}{2p(x)}$, the diffusion coefficient $\sigma(x) = \sqrt{\frac{a(x)}{p(x)}}$, the initial state X_0 has distribution $\Delta(x)$, and $(B_t, t > 0)$ is standard Brownian motion. Obviously, if $a = 1$ and $p = 1$, we revert back to the simpler model (7.2). What makes the solution $g(x;t)$ to (8.1) a plausible kernel density estimator is that $g(x;t)$ is a pdf with the following properties. First, $g(\,\cdot\,;0)$ is identical to the initial condition of (8.1), that is, to the empirical density $\Delta(x)$. This property is possessed by both the Gaussian kernel density estimator (7.1) and the diffusion estimator (7.3). Second, if $p(x)$ is a pdf on $\mathscr{X}$, then

$$\lim_{t\to\infty} g(x;t) = p(x), \quad x \in \mathscr{X}.$$

This property is similar to the property that the kernel (7.6) and the estimator (7.3) converge to the uniform density on $\mathscr{X} \equiv [0,1]$ as $t \to \infty$. In the context of the diffusion process governed by (8.2), p is the limiting and stationary density of the diffusion. Third, similar to the estimator (7.3) and the Gaussian kernel density estimator (7.1), we can write the solution of (8.1) as:

$$g(x;t) = \frac{1}{N}\sum_{i=1}^{N} \kappa(x, X_i; t), \tag{8.3}$$

where for each fixed $y \in \mathscr{X}$ the diffusion kernel κ satisfies the PDE:

$$\left\{ \begin{array}{rcl} \frac{\partial}{\partial t}\kappa(x,y;t) & = & L\kappa(x,y;t), \quad x \in \mathscr{X},\ t > 0 \\ \kappa(x,y;0) & = & \delta(x-y), \quad x \in \mathscr{X}. \end{array} \right. \tag{8.4}$$

In addition, for each fixed $x \in \mathscr{X}$ the kernel κ satisfies the PDE:

$$\left\{ \begin{array}{rcl} \frac{\partial}{\partial t}\kappa(x,y;t) & = & L^*\kappa(x,y;t), \quad y \in \mathscr{X},\ t > 0 \\ \kappa(x,y;0) & = & \delta(x-y), \quad y \in \mathscr{X}, \end{array} \right. \tag{8.5}$$

where L^* is of the form $\frac{1}{2p(y)}\frac{\partial}{\partial y}\Big(a(y)\frac{\partial}{\partial y}(\ \cdot\ )\Big)$; that is, L^* is the adjoint operator of L. Note that L^* is the *infinitesimal generator* of the Itô diffusion process in (8.2). If the set $\mathscr{X}$ has boundaries, we add the Neumann boundary condition

$$\frac{\partial}{\partial x}\left(\frac{\kappa(x,y;t)}{p(x)}\right)\Bigg|_{x\in\partial\mathscr{X}} = 0, \quad \forall t > 0. \tag{8.6}$$

and $\frac{\partial}{\partial y}\kappa(x,y;t)\big|_{y\in\partial\mathscr{X}} = 0$ to (8.4) and (8.5) respectively. These boundary conditions ensure that $g(x;t)$ integrates to unity for all $t \geqslant 0$. The reason that the kernel κ satisfies both PDEs (8.4) and (8.5) is that (8.4) is the Kolmogorov forward equation [20] corresponding to the diffusion process (8.2), and (8.5) is a direct consequence of the Kolmogorov backward equation. We will use the forward and backward equations to derive the asymptotic properties of the diffusion estimator (8.3). Before we proceed with the asymptotic analysis, we illustrate how the model (8.1) possesses adaptive smoothing properties similar to the ones possessed by the adaptive kernel density estimators [1, 30, 31, 60].

Example 8.1.1 Suppose that the initial condition of PDE (8.1) is $\Delta(x)$ with $N = 500,000$ and $X_1, \ldots, X_N$ are independent draws from $f(x) = 1 - \cos(6\pi x),\ x \in [0,1]$. Suppose further that $p(x) = 4(1-x)^3$ and $a(x) = 1$ on $[0,1]$. Figure 8.1 shows the solution of the PDE (8.1) for two values of the bandwidth: $\sqrt{t} = 4 \times 10^{-4}$ (small) and $\sqrt{t} = 0.89$ (large). Since $p(x)$ is the limiting and stationary density of the diffusion process governed by (8.1), the large bandwidth

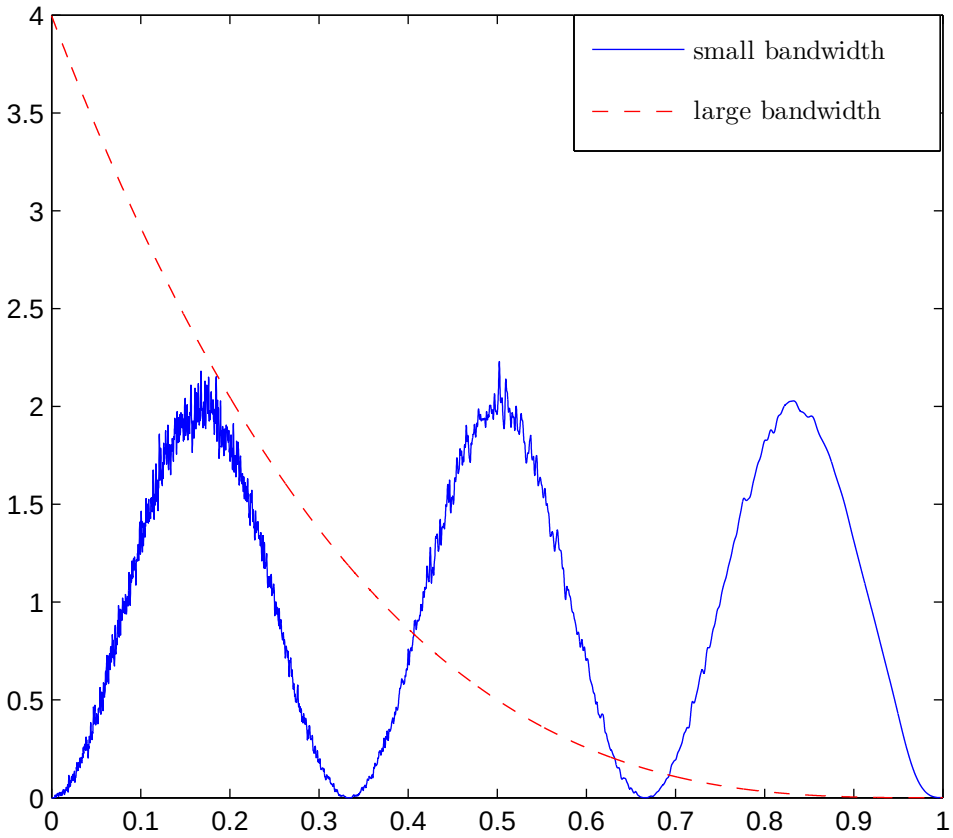

Figure 8.1: Small and large bandwidth behavior of the diffusion density in Example 2.

density is indistinguishable from $p(x)$. The small bandwidth density estimate is much closer to $f(x)$ than to $p(x)$. The crucial feature of the small bandwidth density estimate is that $p(x)$ allows for varying degrees of smoothing across the domain of the data, in particular allowing for greater smoothing to be applied in areas of sparse data, and relatively less in the high density regions. It can be seen from Figure 8.1 that the small time density estimate is noisier in regions where $p(x)$ is large (closer to $x = 0$), and smoother in regions where $p(x)$ is small (closer to $x = 1$). The adaptive smoothing is a consequence of the fact that the diffusion kernel (8.4) has a state-dependent diffusion coefficient $\sigma(x) = \sqrt{a(x)/p(x)}$, which helps diffuse the initial density $\Delta(x)$ at a different rate throughout the state space.

Remark 8.1.1 Even though there is no analytical expression for the diffusion kernel satisfying (8.4), we can write κ in terms of a generalized Fourier series in the case that $\mathscr{X}$ is bounded:

$$\kappa(x, y; t) = p(x) \sum_{k=0}^{\infty} \mathrm{e}^{\lambda_k t} \varphi_k(x) \varphi_k(y), \quad x, y \in [0, 1] \tag{8.7}$$

where $\{\varphi_k\}$ and $\{\lambda_k\}$ are the eigenfunctions and eigenvalues of the Sturm-Liouville problem on $[0, 1]$:

$$\begin{aligned} L^* \varphi_k &= \lambda_k \varphi_k, \quad k = 0, 1, 2, \ldots, \\ \varphi_k'(0) = \varphi_k'(1) &= 0, \quad k = 0, 1, 2, \ldots \end{aligned} \tag{8.8}$$

It is well known (see, e.g., [49]) that $\{\varphi_k\}$ forms a complete orthonormal basis with respect to the weight p for $L^2(0, 1)$. From the expression (8.7) we can see that the kernel satisfies the *detailed balance* equation for a continuous-time Markov process [20]:

$$p(y)\kappa(x, y; t) = p(x)\kappa(y, x; t), \quad \forall t > 0, \quad x, y \in \mathscr{X}. \tag{8.9}$$

The detailed balance equation ensures that the limiting and stationary density of the diffusion estimator (8.3) is $p(x)$. In addition, the kernel satisfies the Chapman-Kolmogorov equation:

$$\int_{\mathscr{X}} \kappa(x_1, x_0; t_1)\, \kappa(x_2, x_1; t_2)\, dx_1 = \kappa(x_2, x_0; t_1 + t_2). \tag{8.10}$$

Note that there is no loss of generality in assuming that the domain is $[0, 1]$, because any bounded domain can be mapped onto $[0, 1]$ by a linear transformation.

Remark 8.1.2 When $p(x)$ is a pdf, an important distance measure between the diffusion estimator (8.3) and $p(x)$ is the divergence measure of Csiszár [17]. The Csiszár distance measure between two continuous probability densities g and p is defined as:

$$\mathcal{D}(g \to p) = \int_{\mathbb{R}} p(x)\ \psi\left(\frac{g(x)}{p(x)}\right) dx,$$

where $\psi : \mathbb{R}_+ \to \mathbb{R}_+$ is a twice continuously differentiable function; $\psi(1) = 0$; and $\psi''(x) > 0$ for all $x \in \mathbb{R}_+$. The diffusion estimator (8.3) possesses the monotonicity property

$$\frac{d}{dt}\mathcal{D}(g \to p) = -\frac{1}{2} \int_{\mathscr{X}} \left(\frac{g(x; t)}{p(x)}\right)^2 \psi''\left(\frac{g(x; t)}{p(x)}\right) dx < 0, \quad g \neq p,\ t > 0.$$

In other words, the distance between the estimator (8.3) and the stationary density p is a

monotonically decreasing function of the bandwidth $\sqrt{t}$. This is why the solution of (8.1) in Figure 8.1 approaches p as the bandwidth becomes larger and larger. Note that Csiszár's family of measures subsumes all of the statistical distance measures used in practice [34, 43]. For example, if $\psi(x) = \frac{x^\alpha - x}{\alpha(\alpha-1)}, \quad \alpha \neq 0, 1$, for some parameter α, then the family of distances indexed by α includes the *Hellinger distance* for $\alpha = 1/2$, *Pearson's* χ^2 *discrepancy measure* for $\alpha = 2$, *Neymann's* χ^2 *measure* for $\alpha = -1$, the *Kullback-Leibler distance* in the limit as $\alpha \to 1$, and *Burg's distance* as $\alpha \to 0$.

8.2 Bias and Variance Analysis

We now examine the asymptotic bias, variance and MISE of the diffusion estimator (8.3). In order to derive the asymptotic properties of the proposed estimator, we need the small bandwidth behavior of the diffusion kernel satisfying (8.4). This is provided by the following lemma.

Lemma 8.2.1 *Assume that the functions $a(x)$ and $p(x)$ are such that,*

$$\begin{aligned} c_1 = \sqrt{\int_{-\infty}^{\infty} \left(\frac{Lq(z)}{q(z)}\right)^2 dz} < \infty, \quad q(z) := \frac{p(z)}{a^{1/4}(z)p^{1/4}(z)}, \\ \lim_{z\to\infty} \int_{z_0}^{z} \sqrt{p(s)/a(s)}\, ds = \infty. \end{aligned} \tag{8.11}$$

Then, the leading small bandwidth asymptotic behavior of the kernel satisfying (8.4) *and* (8.5) *on $\mathscr{X} \equiv \mathbb{R}$ is:*

$$\kappa(x, y; t) \sim \frac{p(x)}{\sqrt{2\pi t}\,[p(x)a(x)a(y)p(y)]^{1/4}} \exp\left\{-\frac{1}{2t}\left[\int_y^x \sqrt{\frac{p(s)}{a(s)}}\, ds\right]^2\right\}, \quad t \downarrow 0.$$

We denote the asymptotic approximation on the right-hand side by $\widetilde{\kappa}(x, y; t)$. Thus, $\kappa(x, y; t) \sim \widetilde{\kappa}(x, y; t)$ as $t \downarrow 0$.

The somewhat lengthy and technical proof is given in Appendix A.2. A few remarks about the technical conditions on a and p now follow. Conditions (8.11) are trivially satisfied if a, p and its derivatives up to order 2 are all bounded from above, and $p(x) \geqslant p_0 > 0$ and $a(x) \geqslant a_0 > 0$. In other words, we can clip $p(x)$ away from zero and use $a(x) = p^\alpha(x)$ for some real constant α, and the conditions (8.11) are satisfied. Such clipping procedures have been applied in the traditional kernel density estimation setting, see [1, 15, 31, 32, 60]. Note the conditions are more easily satisfied when p is heavy-tailed. For example, if $a(x) = p(x)$, then p could be any

regularly varying pdf of the form $p \propto (1+|x|)^{-\alpha}, \alpha > 1$. Lemma 8.2.1 is required for deriving the asymptotic properties of the estimator, all collected in the following theorem.

Theorem 8.2.1 *Let $t = t_N$ be such that $\lim_{N\to\infty} t_N = 0$, $\lim_{N\to\infty} N\sqrt{t_N} = \infty$. Assume that f is twice continuously differentiable and that the domain $\mathscr{X} \equiv \mathbb{R}$. Then:*

1. *The pointwise bias has the asymptotic behavior*

$$\mathbb{E}_f[g(x;t)] - f(x) = t\,Lf(x) + O(t^2), \quad N \to \infty. \tag{8.12}$$

2. *The integrated squared bias has the asymptotic behavior*

$$||\mathbb{E}_f[g(\,\cdot\,;t)] - f||^2 \sim t^2\,||Lf||^2 = \frac{1}{4}t^2||(a\,(f/p)')'||^2, \quad N \to \infty. \tag{8.13}$$

3. *The pointwise variance has the asymptotic behavior*

$$\mathrm{Var}_f[g(x;t)] \sim \frac{f(x)}{2N\sqrt{\pi t}\,\sigma(x)}, \quad N \to \infty, \tag{8.14}$$

   where $\sigma^2(x) = a(x)/p(x)$.

4. *The integrated variance has the asymptotic behavior*

$$\int \mathrm{Var}_f[g(x;t)]\,dx \sim \frac{\mathbb{E}_f[\sigma^{-1}(X)]}{2N\sqrt{\pi t}}, \quad N \to \infty. \tag{8.15}$$

5. *Combining the leading order bias and variance terms gives the asymptotic approximation to the MISE:*

$$\mathrm{AMISE}\{g\}(t) = \frac{1}{4}t^2||(a(f/p)')'||^2 + \frac{\mathbb{E}_f[\sigma^{-1}(X)]}{2N\sqrt{\pi t}}. \tag{8.16}$$

6. *Hence, the square of the asymptotically optimal bandwidth is*

$$t^* = \left(\frac{\mathbb{E}_f[\sigma^{-1}(X)]}{2N\sqrt{\pi}\,||\,Lf\,||^2}\right)^{2/5}, \tag{8.17}$$

   which gives the minimum:

$$\min_t \mathrm{AMISE}\{g\}(t) = N^{-4/5}\,\frac{5\,[\mathbb{E}_f\sigma^{-1}(X)]^{4/5}\,||\,Lf\,||^{2/5}}{2^{14/5}\pi^{2/5}}. \tag{8.18}$$

The proof is given in Appendix A.3.

We make the following observations. First, if $p \not\equiv f$, the rate of convergence of (8.18) is $O(N^{-4/5})$, the same as the rate of the Gaussian kernel density estimator in (A.5). The multiplicative constant of $N^{-4/5}$ in (8.18), however, can be made very small by choosing p to be a *pilot density estimate* of f. Preliminary or pilot density estimates are used in most adaptive kernel methods [79]. Second, if $p \equiv f$, then the leading bias term (8.12) is 0. In fact, if f is infinitely smooth, the pointwise bias is exactly zero, as can be seen from:

$$\mathbb{E}_f[g(x;t)] = \sum_{k=0}^{\infty} \frac{t^k}{k!} L^k f(x), \quad f \in C^\infty,$$

where $L^{n+1} = LL^n$ and L^0 is the identity operator. In addition, if $a = p \propto 1$, then the bias term (8.12) is equivalent to the bias term (A.1) of the Gaussian kernel density estimator. Third, (8.14) suggests that in regions where the pilot density $p(x)$ is large (which is equivalent to small diffusion coefficient $\sigma(x)$) and $f(x)$ is large, the pointwise variance will be large. Conversely, in regions with few observations (that is, where the diffusion coefficient $\sigma(x)$ is high and $f(x)$ is small) the pointwise variance is low. In other words, the ideal variance behavior results when the diffusivity $\sigma(x)$ behaves inversely proportional to $f(x)$.

8.3 Special Cases of the Diffusion Estimator

We shall now show that the diffusion kernel estimator (8.3) is a generalization of some well-known modifications of the Gaussian kernel density estimator (7.1). Examples of modifications and improvements subsumed as special cases of (8.3) are as follows.

1. If $a(x) = p(x) \propto 1$ in (8.3) and $\mathscr{X} \equiv \mathbb{R}$, then the kernel κ reduces to the Gaussian kernel and we obtain (7.1).

2. If $a(x) = 1$ and $p(x) = f_p(x)$, where f_p is a clipped pilot density estimate of f (see [1, 32, 60]), then from Lemma 8.2.1, we have

$$\kappa(x,y;t) \sim \widetilde{\kappa}(x,y;t) = \frac{f_p(x)}{\sqrt{2\pi t}\,(f_p(x)f_p(y))^{1/4}} \exp\left\{-\frac{1}{2t}\left[\int_y^x \sqrt{f_p(s)}\,ds\right]^2\right\}.$$

   Thus, in the neighborhood of y such that $|x-y| = O(t^\beta),\ \beta > 1/3$, we have

$$\kappa(x,y;t) \sim \frac{1}{\sqrt{2\pi t/f_p(x)}} \exp\left\{-\frac{(x-y)^2}{2t/f_p(x)}\right\}, \quad t \downarrow 0.$$

   In other words, in the neighborhood of y, κ is asymptotically equivalent to a Gaussian

kernel with mean y and bandwidth $\sqrt{t/f_p(y)}$, which is precisely the Abramson's variable bandwidth [1] modification as applied to the Gaussian kernel. The Abramson's square root law states that the asymptotically optimal variable bandwidth is proportional to $f_p^{-1/2}(y)$.

3. If we choose $a(x) = p(x) = f_p(x)$, then in an $O(t^\beta)$, $\beta > 0$ neighborhood of y, the kernel $\kappa(x, y; t)$ behaves asymptotically as a Gaussian kernel with location $y + \frac{t}{2}\frac{f_p'(y)}{f_p(y)}$ and bandwidth $\sqrt{t}$:

$$\kappa(x, y; t) \sim \frac{1}{\sqrt{2\pi t}} \exp\left\{-\frac{1}{2t}\left(x - y - \frac{t}{2}\frac{f_p'(y)}{f_p(y)}\right)^2\right\}, \quad t \downarrow 0.$$

This is precisely the data sharpening modification described in [70], where the locations of the data points are shifted prior to the application of the kernel density estimate. Thus, in our paradigm, data sharpening is equivalent to using the diffusion (8.1) with drift $\mu(x) = \frac{f_p'(x)}{2f_p(x)}$ and diffusion coefficient $\sigma(x) = 1$.

4. Finally, if we set $p(x) = f_p(x)$ and $a(x) = p^\alpha(x)$, $\alpha \in [0, 1]$, then we obtain a method that is a combination of both the data sharpening and the variable bandwidth of Abramson. The kernel κ behaves asymptotically (in an $O(t^\beta)$, $\beta > 1/3$ neighborhood of y) like a Gaussian kernel with location $y + t\mu(y) = y + \frac{\alpha t}{2} f_p^{\alpha-2}(y) f_p'(y)$ and bandwidth $\sqrt{t\,\sigma^2(y)} = \sqrt{t\, f_p^{\alpha-1}(y)}$. Similar variable location and scale kernel density estimators are considered in [60].

The proposed method thus unifies many of the already existing ideas for variable scale and location kernel density estimators. One problem remains unresolved — the data-driven selection of the parameter t, which plays the role of a bandwidth. This is the subject of the next chapter.

Chapter 9

Bandwidth Selection Algorithm

In this chapter we introduce a data-driven bandwidth selection rule that builds on the currently existing methods to achieve unparalleled practical performance. We conduct numerical experiments with some well-known density estimation test cases.

9.1 Bandwidth Selection for Gaussian Kernel Estimator

Before we explain how to estimate the bandwidth $\sqrt{t^*}$ in (8.17) of the diffusion estimator (8.3), we explain how to estimate the bandwidth $\sqrt{{}_*t}$ in (A.4) (see Appendix A.1) of the Gaussian kernel density estimator (7.1). Here we present a new plug-in bandwidth selection procedure based on the ideas in [39, 56, 57, 72] to achieve unparalleled practical performance. The highlighting feature of the proposed method is that it does not use normal reference rules and is thus completely data-driven.

It is clear from (A.4) in Appendix A.1 that to compute the optimal ${}_*t$ for the Gaussian kernel density estimator (7.1) one needs to estimate the functional $||f''||^2$. Thus we consider the problem of estimating $||f^{(j)}||^2$ for an arbitrary integer $j \geqslant 1$. The identity $||f^{(j)}||^2 = (-1)^j \mathbb{E}_f[f^{(2j)}(X)]$ suggests two possible plug-in estimators. The first one is:

$$(-1)^j \widehat{\mathbb{E}_f f^{(2j)}} := \frac{(-1)^j}{N} \sum_{k=1}^{N} \widehat{f}^{(2j)}(X_k; t_j) = \frac{(-1)^j}{N^2} \sum_{k=1}^{N} \sum_{m=1}^{N} \phi^{(2j)}(X_k, X_m; t_j), \tag{9.1}$$

where $\widehat{f}$ is the Gaussian kernel density estimator (7.1). The second estimator is:

$$\begin{aligned} \widehat{||f^{(j)}||^2} &:= ||\widehat{f}^{(j)}(\,\cdot\,; t)||^2 \\ &= \frac{1}{N^2} \sum_{k=1}^{N} \sum_{m=1}^{N} \int_{\mathbb{R}} \phi^{(j)}(x, X_k; t_j) \phi^{(j)}(x, X_m; t_j)\, dx \\ &= \frac{(-1)^j}{N^2} \sum_{k=1}^{N} \sum_{m=1}^{N} \phi^{(2j)}(X_k, X_m; 2\, t_j), \end{aligned} \tag{9.2}$$

where the last line is a simplification following easily from the fact that the Gaussian kernel ϕ satisfies the Chapman-Kolmogorov equation (8.10). For a given bandwidth, both estimators $(-1)^j\widehat{\mathbb{E}_f f^{(2j)}}$ and $||\widehat{f^{(j)}}||^2$ aim to estimate the same quantity, namely $||f^{(j)}||^2$. We select t_j so that both estimators (9.1) and (9.2) are asymptotically equivalent in the mean square error sense. In other words, we choose $t_j = {}_*t_j$ so that both $(-1)^j\widehat{\mathbb{E}_f f^{(2j)}}$ and $||\widehat{f^{(j)}}||^2$ have equal asymptotic mean square error. This gives the following proposition.

Proposition 9.1.1 *The estimators* $(-1)^j\widehat{\mathbb{E}_f f^{(2j)}}$ *and* $||\widehat{f^{(j)}}||^2$ *have the same asymptotic mean square error when*

$$ {}_*t_j = \left(\frac{1 + \frac{1}{2^{j+1/2}}}{3} \, \frac{1 \times 3 \times 5 \times \cdots \times (2j-1)}{N\sqrt{\pi/2}\, ||f^{(j+1)}||^2} \right)^{\frac{2}{3+2j}} . \tag{9.3} $$

Proof: The arguments are similar to the ones used in [79]. Under the assumptions that t_j depends on N such that $\lim_{N\to\infty} t_j = 0$ and $\lim_{N\to\infty} N\, t_j^{j+1/2} = \infty$, we can take the expectation of the estimator (9.1) and obtain the expansion ($t_j = t$):

$$ \begin{aligned} &\mathbb{E}_f\left[\widehat{\mathbb{E}_f f^{(2j)}}\right] \\ &= \frac{1}{N}\phi^{(2j)}(0,0;t) + \frac{N-1}{N}\iint f(x)f(y)\phi^{(2j)}(x,y;t)\,dxdy \\ &= -\frac{1 \times 3 \times \cdots \times (2j-1)}{t^{j+1/2}\sqrt{2\pi}N} + \int f(x)\left(f^{(2j)}(x) + \frac{t}{2}f^{2(j+1)}(x) + o(t)\right)dx + O(N^{-1}) \\ &= -\frac{1 \times 3 \times 5 \times \cdots \times (2j-1)}{t^{j+1/2}\sqrt{2\pi}N} + \frac{t}{2}||f^{(j+1)}||^2 + (-1)^j||f^{(j)}||^2 + O(N^{-1}), \quad N \to \infty. \end{aligned} $$

Hence, the squared bias has asymptotic behavior ($N \to \infty$):

$$ \left((-1)^j\mathbb{E}_f\left[\widehat{\mathbb{E}_f f^{(2j)}}\right] - ||f^{(j)}||^2\right)^2 \sim \left(\frac{1 \times 3 \times \cdots \times (2j-1)}{t^{j+1/2}\sqrt{2\pi}N} - \frac{t}{2}||f^{(j+1)}||^2\right)^2 . $$

A similar argument (see [79]) shows that the variance is of the order $O(N^{-2}t^{-2j-1/2})$, which is of lesser order than the squared bias. This implies that the leading order term in the asymptotic mean square error of $\widehat{\mathbb{E}_f f^{(2j)}}$ is given by the asymptotic squared bias. There is no need to derive the asymptotic expansion of $\mathbb{E}_f\left[||\widehat{f^{(j)}}||^2\right]$, because inspection of (9.2) and (9.1) shows that $||\widehat{f^{(j)}}||^2$ exactly equals $(-1)^j\widehat{\mathbb{E}_f f^{(2j)}}$ when the latter is evaluated at $2\,t_j$. In other words,

$$ (-1)^j\mathbb{E}_f\left[||\widehat{f^{(j)}}||^2\right] = -\frac{1 \times 3 \times 5 \times \cdots \times (2j-1)}{(2t)^{j+1/2}\sqrt{2\pi}N} + t||f^{(j+1)}||^2 + O(1+N^{-1}). $$

Again, the leading term of the asymptotic mean square error of $||\widehat{f^{(j)}}||^2$ is given by the leading

term of the squared bias of $\widehat{||f^{(j)}||^2}$. Thus equalizing the asymptotic mean squared error of both estimators is the same as equalizing their respective asymptotic squared biases. This yields the equation:

$$\left(\frac{1\times 3\times\cdots\times(2j-1)}{(2t)^{j+1/2}\sqrt{2\pi}N}-t||f^{(j+1)}||^2\right)^2=\left(\frac{1\times 3\times\cdots\times(2j-1)}{t^{j+1/2}\sqrt{2\pi}N}-\frac{t}{2}||f^{(j+1)}||^2\right)^2.$$

The positive solution of the equation yields the desired ${}_*t_j$. □

Thus, for example,

$${}_*t_2=\left(\frac{8+\sqrt{2}}{24}\,\frac{3}{N\sqrt{\pi/2}\,||f^{(3)}||^2}\right)^{2/7} \tag{9.4}$$

is our bandwidth choice for the estimation of $||f''||^2$. We estimate each ${}_*t_j$ by

$${}_*\widehat{t}_j=\left(\frac{1+\frac{1}{2^{j+1/2}}}{3}\,\frac{1\times 3\times 5\times\cdots\times(2j-1)}{N\sqrt{\pi/2}\,\widehat{||f^{(j+1)}||^2}}\right)^{\frac{2}{3+2j}}. \tag{9.5}$$

Computation of $\widehat{||f^{(j+1)}||^2}$ requires estimation of ${}_*t_{j+1}$ itself, which in its turn requires estimation of ${}_*t_{j+2}$ and so on, as seen from formulas (9.2) and (9.5). We are faced with the problem of estimating the infinite sequence $\{{}_*t_{j+k}, k\geqslant 1\}$. It is clear, however, that given ${}_*t_{l+1}$ for some $l>0$ we can estimate all $\{{}_*t_j,\ 1\leqslant j\leqslant l\}$ recursively, and then estimate ${}_*t$ itself from (A.4). This motivates the *l-stage direct plug-in bandwidth selector* [39, 72, 79], defined as follows.

1. For a given integer $l>0$, estimate ${}_*t_{l+1}$ via (9.3) and $||f^{(l+2)}||^2$ computed by assuming that f is a normal density with mean and variance estimated from the data. Denote the estimate by ${}_*\widehat{t}_{l+1}$.

2. Use ${}_*\widehat{t}_{l+1}$ to estimate $||f^{(l+1)}||^2$ via the plug-in estimator (9.2) and ${}_*\widehat{t}_l$ via (9.5). Then use ${}_*\widehat{t}_l$ to estimate ${}_*\widehat{t}_{l-1}$ and so on until we obtain an estimate of ${}_*\widehat{t}_2$.

3. Use the estimate of ${}_*\widehat{t}_2$ to compute ${}_*\widehat{t}$ from (A.4).

The l-stage direct plug-in bandwidth selector thus involves the estimation of l functionals $\{||f^{(j)}||, 2\leqslant j\leqslant l+1\}$ via the plug-in estimator (9.2). We can describe the procedure in a more abstract way as follows. Denote the functional dependence of ${}_*\widehat{t}_j$ on ${}_*\widehat{t}_{j+1}$ in formula (9.5) as

$${}_*\widehat{t}_j=\gamma({}_*\widehat{t}_{j+1}).$$

It is then clear that ${}_*\widehat{t}_j=\gamma(\gamma({}_*\widehat{t}_{j+2}))=\gamma(\gamma(\gamma({}_*\widehat{t}_{j+3})))=\cdots$. For simplicity of notation we

define the composition

$$\gamma^{[k]}(t) = \underbrace{\gamma(\cdots\gamma(\gamma}_{k \text{ times}}(t))\cdots), \quad k \geqslant 1,$$

where $\gamma^{[1]}(t) = \gamma(t)$. Then, ${}_*\widehat{t}_j = \gamma^{[k]}({}_*\widehat{t}_{j+k})$ or alternatively ${}_*\widehat{t}_{j-k} = \gamma^{[k]}({}_*\widehat{t}_j)$ for $j \geqslant k \geqslant 1$. Inspection of formulas (9.5) and (A.4) shows that the estimate of ${}_*t$ satisfies

$${}_*\widehat{t} = \xi\ {}_*\widehat{t}_1 = \xi\,\gamma({}_*\widehat{t}_2) = \xi\,\gamma^{[2]}({}_*\widehat{t}_3) = \cdots = \xi\,\gamma^{[l]}({}_*\widehat{t}_{1+l}), \quad \xi = \left(\frac{6\sqrt{2}-3}{7}\right)^{2/5} \approx 0.90.$$

Then, for a given integer $l > 0$, the l-stage direct plug-in bandwidth selector consists of computing

$${}_*\widehat{t} = \xi\,\gamma^{[l]}({}_*t_{l+1}),$$

where ${}_*t_{l+1}$ is estimated via (9.3) by assuming that f in $||f^{(l+2)}||^2$ is a normal density with mean and variance estimated from the data. The weakest point of this procedure is that we assume that the true f is a Gaussian density in order to compute $||f^{(l+2)}||^2$. This assumption can lead to arbitrarily bad estimates of ${}_*t$, when for example the true f is far from being Gaussian. Instead, we propose to find a solution to the nonlinear equation

$$t = \xi\ \gamma^{[l]}(t), \tag{9.6}$$

for some l, using either fixed point iteration or Newton's method with initial guess $t = 0$. The fixed point iteration version is formalized in the following algorithm.

Algorithm 9.1.1 *Given $l > 2$, execute the following steps.*

1. *Initialize with $z_0 = \varepsilon$, where ε is machine precision, and $n = 0$;*
2. *Set $z_{n+1} = \xi\,\gamma^{[l]}(z_n)$.*
3. *If $|z_{n+1} - z_n| < \varepsilon$, stop and set ${}_*\widehat{t} = z_{n+1}$; otherwise, set $n := n + 1$ and repeat from Step 2.*
4. *Deliver the Gaussian kernel density estimator (7.1) evaluated at ${}_*\widehat{t}$ as the final estimator of f, and ${}_*\widehat{t}_2 = \gamma^{[l-1]}(z_{n+1})$ as the bandwidth for the optimal estimation of $||f''||^2$.*

Numerical experience suggests the following. First, the fixed-point algorithm does not fail to find a root of the equation $t = \xi\,\gamma^{[l]}(t)$. Second, the root appears to be unique. Third, the solutions to the equations

$$t = \xi\,\gamma^{[5]}(t)$$

and

$$t = \xi\, \gamma^{[l+5]}(t),$$

for any $l > 0$ do not differ in any practically meaningful way. In other words, there were no gains to be had by increasing the stages of the bandwidth selection rule beyond $l = 5$. We recommend setting $l = 5$. Lastly, the numerical procedure for the computation of $\gamma^{[5]}(t)$ is fast when implemented using the Discrete Cosine Transform [4].

The plug-in method described in Algorithm 9.1.1 has superior practical performance compared to existing plug-in implementations, including the particular *solve-the-equation* rule of Sheather and Jones [72, 79]. To illustrate the significant improvement of the plug-in method in Algorithm 9.1.1, consider, for example, the case where f is a mixture of two Gaussian densities with a common variance of 1 and means of -30 and 30.

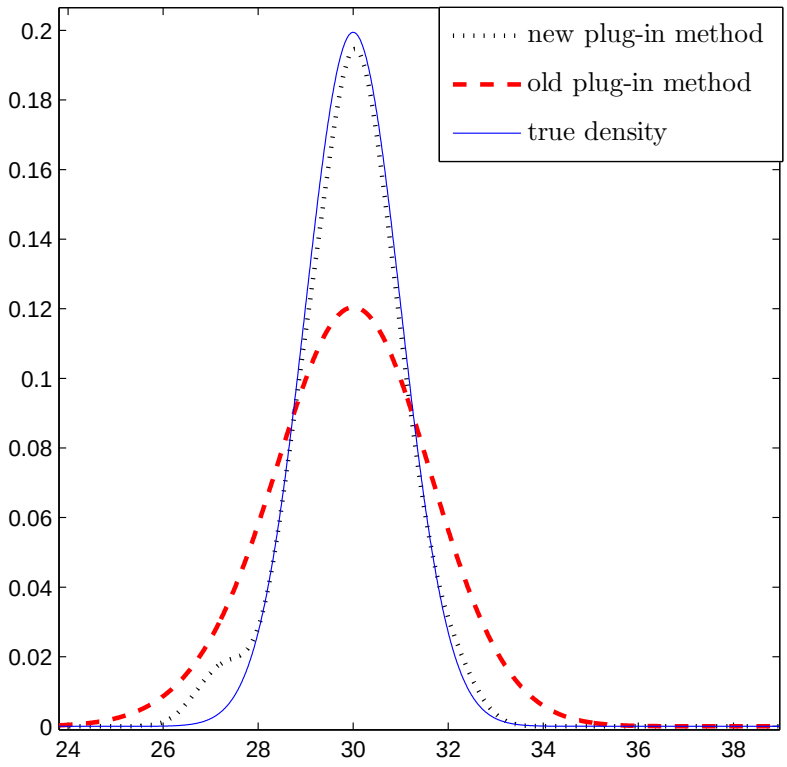

Figure 9.1: The new bandwidth selection rule in Algorithm 9.1.1 leads to improved performance compared to old plug-in rules.

Figure 9.1 shows the right mode of f, and the two estimates resulting from the old plug-in rule [72] and the plug-in rule of Algorithm 9.1.1. The left mode is not displayed, but looks

similar. The integrated squared error using the new plug-in bandwidth estimate, $||f-\widehat{f}(\,\cdot\,;{}_*\widehat{t})||^2$, is one 10-th of the error using the old bandwidth selection rule. This examples demonstrates that the new bandwidth selection procedure passes the *bi-modality test* [18], which consists of testing the performance of a bandwidth selection procedure using a bimodal target density, with the two modes at some distance from each other. It has been demonstrated in [18] that by separating the modes of the target density enough, existing plug-in selection procedures can be made to perform arbitrarily poorly due to the adverse effects of the normal reference rules. The proposed plug-in method in Algorithm 9.1.1 performs much better, because it uses the theoretical ideas developed in existing plug-in rules [72], except for the detrimental normal reference rules. A Matlab implementation of Algorithm 9.1.1 is freely available from [4], and includes other examples of improved performance.

Algorithm 9.1.1 can be extended to bandwidth selection in higher dimensions. For completeness we describe the two-dimensional version of the algorithm in the next section.

9.2 Bandwidth Selection in Higher Dimensions

Algorithm 9.1.1 can be extended to two dimensions for the estimation of a pdf $f(\mathbf{x})$ on $\mathbb{R}^2$. Assuming a Gaussian kernel

$$\phi(\mathbf{x},\mathbf{y};t) = \frac{1}{2\pi t}\,\mathrm{e}^{\frac{(\mathbf{x}-\mathbf{y})^T(\mathbf{x}-\mathbf{y})}{2t}},$$

where $\mathbf{x} = [x_1, x_2]^T$ and $\mathbf{y} = [y_1, y_2]^T$, the asymptotically optimal squared bandwidth is given by ([79], p. 99)

$$t^* = \Big(2\pi N\Big(\psi_{0,2} + \psi_{2,0} + 2\psi_{1,1}\Big)\Big)^{-1/3},$$

where

$$\begin{aligned}\psi_{i,j} &= (-1)^{i+j}\int_{\mathbb{R}^2} f(\mathbf{x})\,\frac{\partial^{2(i+j)}}{\partial x_1^{2i}\partial x_2^{2j}} f(\mathbf{x})\,d\mathbf{x}, \quad i,j\in\mathbb{N}^+,\\ &= \int\left(\frac{\partial^{(i+j)}}{\partial x_1^{i}\partial x_2^{j}} f(\mathbf{x})\right)^2 d\mathbf{x}.\end{aligned} \tag{9.7}$$

Note that our definition of ψ differs slightly from the definition of ψ in [79]. Here the partial derivatives under the integral sign are applied $2(i+j)$ times, while in [79] they are applied $(i+j)$ times. Similar to the one dimensional case, there are two viable plug-in estimators for $\psi_{i,j}$. The first one is derived from the first line of (9.7):

$$\widetilde{\psi}_{i,j} = \frac{(-1)^{i+j}}{N^2}\sum_{k=1}^{N}\sum_{m=1}^{N}\frac{\partial^{2(i+j)}}{\partial x_1^{2i}\partial x_2^{2j}}\phi(\mathbf{X}_m,\mathbf{X}_k;t_{i,j}), \tag{9.8}$$

and the second one is derived from the second line of (9.7):

$$\begin{aligned}\widehat{\psi}_{i,j} &= \frac{1}{N^2}\sum_{k=1}^{N}\sum_{m=1}^{N}\int \frac{\partial^{(i+j)}}{\partial x_1^i \partial x_2^j}\phi(\mathbf{x},\mathbf{X}_m;t_{i,j})\frac{\partial^{(i+j)}}{\partial x_1^i \partial x_2^j}\phi(\mathbf{x},\mathbf{X}_k;t_{i,j})\,d\mathbf{x} \\ &= \frac{(-1)^{i+j}}{N^2}\sum_{k=1}^{N}\sum_{m=1}^{N}\frac{\partial^{2(i+j)}}{\partial x_1^{2i}\partial x_2^{2j}}\phi(\mathbf{X}_m,\mathbf{X}_k;2\,t_{i,j}).\end{aligned} \tag{9.9}$$

The asymptotic expansion of the squared bias of estimator $\widetilde{\psi}_{i,j}$ is given by ([79], p. 113):

$$\left(\mathbb{E}_f[\widetilde{\psi}_{i,j}]-\psi_{i,j}\right)^2 \sim \left(\frac{q(i)\,q(j)}{N\,t_{i,j}^{i+j+1}}+\frac{t_{i,j}}{2}\left(\psi_{i+1,j}+\psi_{i,j+1}\right)\right)^2, \quad N\to\infty, \tag{9.10}$$

where

$$q(j)=\begin{cases}(-1)^j\,\frac{1\times3\times5\times\cdots\times(2j-1)}{\sqrt{2\pi}} & j\geqslant 1\\ \frac{1}{\sqrt{2\pi}} & j=0\end{cases}.$$

Thus, we have:

$$\left(\mathbb{E}_f[\widehat{\psi}_{i,j}]-\psi_{i,j}\right)^2 \sim \left(\frac{q(i)\,q(j)}{N\,(2\,t_{i,j})^{i+j+1}}+t_{i,j}\left(\psi_{i+1,j}+\psi_{i,j+1}\right)\right)^2, \quad N\to\infty. \tag{9.11}$$

For both estimators the squared bias is the dominant term in the asymptotic mean squared error, because the variance is of the order $O(N^{-2}t^{-2i-2j-1})$. It follows that both estimators will have the same leading asymptotic mean square error term provided that

$$t_{i,j}=\left(\frac{1+2^{-i-j-1}}{3}\,\frac{-2\,q(i)\,q(j)}{N(\psi_{i+1,j}+\psi_{i,j+1})}\right)^{1/(2+i+j)}. \tag{9.12}$$

We estimate $t_{i,j}$ via

$$\widehat{t}_{i,j}=\left(\frac{1+2^{-i-j-1}}{3}\,\frac{-2\,q(i)\,q(j)}{N(\widehat{\psi}_{i+1,j}+\widehat{\psi}_{i,j+1})}\right)^{1/(2+i+j)}. \tag{9.13}$$

Thus, estimation of $\psi_{i,j}$ requires estimation of $\psi_{i,j+1}$ and $\psi_{i+1,j}$, which in turn requires estimation of $\psi_{i+2,j}$, $\psi_{i+1,j+1}$, $\psi_{i,j+2}$ and so on applying formula (9.13) recursively. Observe that to estimate all $\psi_{i,j}$ for which $i+j=k$, that is, $\{\psi_{i,j} : i+j=k\}$, we need estimates of all $\{\psi_{i,j} : i+j=k+1\}$. For example, from formula (9.13) we can see that estimation of $t_{2,0}$, $t_{1,1}$, $t_{0,2}$ requires estimation of $t_{3,0}$, $t_{2,1}$, $t_{1,2}$, $t_{0,3}$.

For a given integer $k\geqslant 3$, we define the function $\gamma(t)$ as follows. Given an input $t>0$,

1. Set $\widehat{t}_{i,j}=t$ for all $i+j=k$.

2. Use the set $\{\widehat{t}_{i,j} : i+j=k\}$ to compute all functionals $\{\widehat{\psi}_{i,j} : i+j=k\}$ via (9.9).

3. Use $\{\widehat{\psi}_{i,j} : i+j=k\}$ to compute $\{\widehat{t}_{i,j} : i+j=k-1\}$ via (9.13).

4. If $k=2$ go to Step 5; otherwise set $k := k-1$ and repeat from Step 2.

5. Use $\{\widehat{\psi}_{i,j} : i+j=2\}$ to output

$$\gamma(t) = \left(2\pi N\left(\widehat{\psi}_{0,2} + \widehat{\psi}_{2,0} + 2\widehat{\psi}_{1,1}\right)\right)^{-1/3}.$$

The bandwidth selection rule simply consists of solving the equation $\gamma(t) = t$ for a given $k \geqslant 3$ via either the fixed point iteration in Algorithm 9.1.1 (ignoring Step 4) or by using Newton's method. We obtain excellent numerical results for $k = 4$ or $k = 5$. Higher values of k did not change the value of t in any significant way, but only increased the computational cost of evaluating the function $\gamma(t)$. Again note that this appears to be the first successful plug-in bandwidth selection rule that does not involve any arbitrary reference rules, but it purely data-driven. An efficient Matlab implementation of the bandwidth selection rule described here, and using the two dimensional Discrete Cosine Transform, can be downloaded freely from [4]. The Matlab implementation takes an additional step in which, once a fixed point of $\gamma(t)$ has been found, the final set of estimates $\{\widehat{\psi}_{i,j} : i+j=2\}$ is used to compute the entries $\sqrt{t_{X_1}}$ and $\sqrt{t_{X_2}}$ of the optimal diagonal bandwidth matrix ([79], p. 111) for a Gaussian kernel of the form

$$\frac{1}{2\pi\sqrt{t_{X_1}\, t_{X_2}}}\, \mathrm{e}^{-\frac{(x_1-y_1)^2}{2t_{X_1}} - \frac{(x_2-y_2)^2}{2t_{X_2}}}.$$

These entries are estimated via the formulas:

$$t_{X_1} = \left(\frac{\widehat{\psi}_{0,2}^{3/4}}{4\pi\, N\, \widehat{\psi}_{2,0}^{3/4}\left(\widehat{\psi}_{1,1} + \sqrt{\widehat{\psi}_{2,0}\,\widehat{\psi}_{0,2}}\right)}\right)^{1/3},$$

and

$$t_{X_2} = \left(\frac{\widehat{\psi}_{2,0}^{3/4}}{4\pi\, N\, \widehat{\psi}_{0,2}^{3/4}\left(\widehat{\psi}_{1,1} + \sqrt{\widehat{\psi}_{2,0}\,\widehat{\psi}_{0,2}}\right)}\right)^{1/3}.$$

9.3 Bandwidth Selection for the Diffusion Estimator

We now discuss the bandwidth choice for the diffusion estimator (8.3). In the following argument we assume that f is as many times continuously differentiable as needed. Computation of t^* in (8.17) requires an estimate of $||Lf||^2$ and $\mathbb{E}_f[\sigma^{-1}(X)]$. We estimate $\mathbb{E}_f[\sigma^{-1}(X)]$ via the unbiased estimator $\frac{1}{N}\sum_{i=1}^N \sigma^{-1}(X_i)$. The identity $||Lf||^2 = \mathbb{E}_f L^*Lf(X)$ suggests two possible plug-in estimators. The first one is:

$$\begin{aligned}\widehat{\mathbb{E}_f L^*Lf} :&= \frac{1}{N}\sum_{j=1}^N L^*Lg(x;t_2)\Big|_{x=X_j} \\ &= \frac{1}{N^2}\sum_{i=1}^N\sum_{j=1}^N L^*L\kappa(x,X_i;t_2)\Big|_{x=X_j},\end{aligned} \tag{9.14}$$

where $g(x;t_2)$ is the diffusion estimator (8.3) evaluated at t_2, and $\mathscr{X} \equiv \mathbb{R}$. The second estimator is:

$$\begin{aligned}\widehat{||Lf||^2} :&= ||Lg(\,\cdot\,;t_2)||^2 = \Big|\Big|\frac{\partial g}{\partial t}(\,\cdot\,;t_2)\Big|\Big|^2 \\ &= \frac{1}{N^2}\sum_{i=1}^N\sum_{j=1}^N \int_{\mathbb{R}} \frac{\partial\kappa}{\partial t}(x,X_i;t_2)\frac{\partial\kappa}{\partial t}(x,X_j;t_2)\,dx \\ &= \frac{1}{N^2}\sum_{i=1}^N\sum_{j=1}^N L^*L\kappa(x,X_i;2\,t_2)\Big|_{x=X_j},\end{aligned} \tag{9.15}$$

where the last line is a simplification that follows from the Chapman-Kolmogorov equation (8.10). The optimal t_2^* is derived in the same way that ${}_*t_2$ is derived for the Gaussian kernel density estimator. That is, t_2^* is such that both estimators $\widehat{\mathbb{E}_f L^*Lf}$ and $\widehat{||Lf||^2}$ have the same asymptotic mean square error. This leads to the following proposition.

Proposition 9.3.1 *The estimators $\widehat{\mathbb{E}_f L^*Lf}$ and $\widehat{||Lf||^2}$ have the same asymptotic mean square error when*

$$t_2^* = \left(\frac{8+\sqrt{2}}{24}\,\frac{-3\sqrt{2}\,\mathbb{E}_f[\sigma^{-1}(X)]}{8\sqrt{\pi}\,N\,\mathbb{E}_f[L^*L^2f(X)]}\right)^{2/7}. \tag{9.16}$$

Proof: Although the relevant calculations are lengthier, the arguments here are exactly the same as the ones used in Proposition 1. In particular, we have the same assumptions on t about its dependence on N. For simplicity of notation, the operators L^* and L are here assumed to

apply to the first argument of the kernel κ:

$$\begin{aligned}
&\mathbb{E}_f\left[\widehat{\mathbb{E}_f L^*Lf}\right] = \\
&= \mathbb{E}_f \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} L^*L\kappa(x,X_i;t)\Big|_{x=X_j} \\
&= \frac{1}{N}\int f(x)L^*L\kappa(x,X_i;t)\Big|_{X_i=x}dx + \frac{N-1}{N}\iint f(y)f(x)L^*L\kappa(x,y;t)dydx \\
&= \frac{3\sqrt{2}\,\mathbb{E}_f[\sigma^{-1}(X)]}{8\sqrt{\pi}\,t^{5/2}N} + O(N^{-1}t^{-3/2}) + \iint f(y)f(x)L^*L\kappa(x,y;t)dydx + O(N^{-1}) \\
&= \frac{3\sqrt{2}\,\mathbb{E}_f[\sigma^{-1}(X)]}{8\sqrt{\pi}\,t^{5/2}N} + \int f(y)\int L^*Lf(x)\kappa(x,y;t)\,dx\,dy + O(N^{-1}(1+t^{-3/2})) \\
&= \frac{3\sqrt{2}\,\mathbb{E}_f[\sigma^{-1}(X)]}{8\sqrt{\pi}\,t^{5/2}N} + ||Lf||^2 + t\int f(y)\,L^*L^2f(y)\,dy + O(N^{-1}(1+t^{-3/2}) + t^2),
\end{aligned}$$

where we have used a consequence of Lemma 8.2.1:

$$\int f(x)\; L^*L\kappa(x,X_i;t)\Big|_{X_i=x}dx \sim \frac{3\sqrt{2}\,\mathbb{E}_f[\sigma^{-1}(X)]}{8\sqrt{\pi}\,t^{5/2}}, \quad t\downarrow 0,$$

and a consequence of the detailed balance equation (8.9):

$$\begin{aligned}
\int L^*Lf(x)\kappa(x,y;t)\,dx &= \int \frac{p(x)L^*Lf(x)}{p(y)}\kappa(y,x;t)\,dx \\
&= L^*Lf(y) + t\,L^*L^*Lf(y) + O(t^2).
\end{aligned}$$

Therefore, the squared bias has asymptotic behavior ($N\to\infty$):

$$\left(\mathbb{E}_f\left[\widehat{\mathbb{E}_f L^*Lf}\right] - ||Lf||^2\right)^2 \sim \left(\frac{3\sqrt{2}\,\mathbb{E}_f[\sigma^{-1}(X)]}{8\sqrt{\pi}\,t^{5/2}N} + t\int f(y)\,L^*L^2f(y)\,dy\right)^2.$$

Since estimator $\widehat{||Lf||^2}$ equals $\widehat{\mathbb{E}_f L^*Lf}$ when the latter is evaluated at $2\,t_2$, the asymptotic squared bias of $\widehat{||Lf||^2}$ follows immediately, and we simply repeat the arguments in the proof of Proposition 1 to obtain the desired t_2^* . □

Note that t_2^* has the same rate of convergence to 0 as ${}_*t_2$ in (9.4). In fact, since the Gaussian kernel density estimator is a special case of the diffusion estimator (8.3) when $p(x) = a(x) = 1$, the plug-in estimator (9.15) for the estimation of $||Lf||^2$ reduces to the plug-in estimator for the estimation of $\frac{1}{4}||f''||^2$. In addition, when $p(x) = a(x) = 1$, the t_2^* in (9.16) and ${}_*t_2$ in (9.4) are identical. We thus suggest the following bandwidth selection and estimation procedure for the diffusion estimator (8.3).

Algorithm 9.3.1

1. *Given the data $X_1, \ldots, X_N$, run Algorithm 9.1.1 to obtain the Gaussian kernel density estimator* (7.1) *evaluated at $_*\widehat{t}$ and the optimal bandwidth $\sqrt{_*\widehat{t}_2}$ for the estimation of $||f''||^2$. This is the pilot estimation step.*

2. *Let $p(x)$ be the Gaussian kernel density estimator from Step 1, and let $a(x) = p^\alpha(x)$ for some $\alpha \in [0, 1]$.*

3. *Estimate $||Lf||^2$ via the plug-in estimator* (9.15) *using $\widehat{t}_2^* = {}_*\widehat{t}_2$, where $_*\widehat{t}_2$ is computed in Step 1.*

4. *Substitute the estimate of $||Lf||^2$ into* (8.17) *to obtain an estimate for t^*.*

5. *Deliver the diffusion estimator* (8.3) *evaluated at $\widehat{t}^*$ as the final density estimate.*

The bandwidth selection rule that we use for the diffusion estimator in Algorithm 9.3.1 is a single stage direct plug-in bandwidth selector, where the bandwidth t_2^* for the estimation of the functional $||Lf||^2$ is approximated by $_*\widehat{t}_2$ (which is computed in Algorithm 9.1.1), instead of being derived from a normal reference rule.

In the next section we illustrate the performance of Algorithm 9.3.1 using some well-known test cases for density estimation.

Remark 9.3.1 (Alternative bandwidth selector) An alternative bandwidth selection procedure for the diffusion estimator is described in [5]. The idea is to choose t^* so that the asymptotic variance of the diffusion estimator (8.15) equals the asymptotic variance of the Gaussian kernel density estimator (A.2), that is, $\widehat{t}^* = [\mathbb{E}_f \sigma^{-1}(X)]^2 \, {}_*\widehat{t}$. In this way the difference between the AMISE (8.18) of the diffusion estimator and the AMISE (A.3) of the Gaussian kernel density estimator is solely due to the difference between the asymptotic squared bias of the estimators.

Remark 9.3.2 (Random variable generation) For applications of kernel density estimation, such as the smoothed bootstrap, efficient random variable generation from the diffusion estimator (8.3) is accomplished via the Euler method as applied to the stochastic differential equation (8.2) (see [46]).

Algorithm 9.3.2

1. *Subdivide the interval $[0, \widehat{t}^*]$ into n equal intervals of length $\delta t = \widehat{t}^*/n$ for some large n.*

2. *Generate a random integer I from 1 to N uniformly.*

3. For $i = 1, \ldots, n$, repeat

$$Y_i = Y_{i-1} + \mu(Y_{i-1})\,\delta t + \sigma(Y_{i-1})\sqrt{\delta t}\,Z_i,$$

where $Z_1, \ldots, Z_n \sim_{iid} \mathsf{N}(0,1)$, and $Y_0 = X_I$.

4. Output Y_n as a random variable with approximate density (8.3).

Note that since we are only interested in the approximation of the statistical properties of Y_n, there are no gains to be had from using the more complex Milstein stochastic integration procedure [46].

9.4 Numerical Experiments

In this section we provide a simulation study of the diffusion estimator. In implementing Algorithm 9.3.1, there are a number of issues to consider. First, the numerical solution of the PDE (8.1) is a straightforward application of either finite difference or spectral methods [49]. A Matlab implementation using finite differences and the stiff ODE solver `ode15s.m` is available from the first author upon request. Second, we compute $||Lg(\,\cdot\,;\widehat{t}_2^{\,*})||^2$ in Algorithm 9.3.1 using the approximation:

$$||Lg(\,\cdot\,;t)||^2 = \left|\left|\frac{\partial g}{\partial t}(\,\cdot\,;t)\right|\right|^2 \approx ||g(\,\cdot\,;t+\varepsilon) - g(\,\cdot\,;t)||^2/\varepsilon^2, \quad \varepsilon \ll 1,$$

where $g(\,\cdot\,;t)$ and $g(\,\cdot\,;t+\varepsilon)$ are the successive output of the numerical integration routine (`ode15s.m` in our case). Finally, we selected $\alpha = 1$ or $a(x) = p(x)$ in Algorithm 9.3.1 without using any clipping of the pilot estimate. For a small simulation study with $\alpha = 0$ see [5].

We would like to point out that simulation studies of existing variable-location scale estimators [60, 70, 75] are implemented assuming that the target pdf f and any functionals of f are known *exactly* and no pilot estimation step is employed. In addition, in these simulation studies the bandwidth is chosen so that it is the global minimizer of the exact MISE. Since in practical applications the MISE and all functionals of f are not available, but have to be estimated, we proceed differently in our simulation study. We compare the estimator of Algorithm 9.3.1 with the Gaussian kernel density estimator (7.1), where ${}_*t$ and ${}_*t_2$ are estimated using the new bandwidth selection procedure in Algorithm 9.1.1. Our aim is to assess the benefits of the diffusion estimator (8.3) compared to the Gaussian kernel density estimator (7.1), given that both use the same bandwidth selection procedure and the target pdf f is unknown. We use the bandwidth selection procedure of Algorithm 9.1.1 because of its superiority over existing bandwidth selection procedures, but the results are similar if we use any one of the currently existing

Case	target density $f(x)$	N	Ratio
1	$\frac{1}{2}\mathsf{N}\left(0, \left(\frac{1}{10}\right)^2\right) + \frac{1}{2}\mathsf{N}(5,1)$	100 10^5	0.75 0.85
2	$\frac{2}{3}\mathsf{N}(0,1) + \frac{1}{3}\mathsf{N}\left(0, \left(\frac{1}{10}\right)^2\right)$	300 3×10^5	0.77 0.84
3	$\frac{1}{2}\mathsf{N}(0,1) + \sum_{k=0}^{4} \frac{1}{10}\mathsf{N}\left(\frac{k}{2} - 1, \left(\frac{1}{10}\right)^2\right)$	300 3×10^5	0.78 0.88
4	$\sum_{k=0}^{7} \frac{1}{8}\mathsf{N}\left(3\left(\left(\frac{2}{3}\right)^k - 1\right), \left(\frac{2}{3}\right)^{2k}\right)$	100 10^5	0.87 0.82
5	$\frac{49}{100}\mathsf{N}\left(-1, \left(\frac{2}{3}\right)^2\right) + \frac{49}{100}\mathsf{N}\left(1, \left(\frac{2}{3}\right)^2\right) + \frac{1}{350}\sum_{k=0}^{6} \mathsf{N}\left(\frac{k-3}{2}, \left(\frac{1}{100}\right)^2\right)$	10^3 10^6	0.94 0.85
6	$\frac{2}{7}\sum_{k=0}^{2} \mathsf{N}\left(\frac{12k-15}{7}, \left(\frac{2}{7}\right)^2\right) + \frac{1}{21}\sum_{k=8}^{10} \mathsf{N}\left(\frac{2k}{7}, \left(\frac{1}{21}\right)^2\right)$	10^3 10^6	0.88 0.83
7	$\frac{46}{100}\sum_{k=0}^{1} \mathsf{N}\left(2k-1, \left(\frac{2}{3}\right)^2\right) + \sum_{k=1}^{3} \frac{1}{300}\mathsf{N}\left(-\frac{k}{2}, \left(\frac{1}{100}\right)^2\right) + \sum_{k=1}^{3} \frac{7}{300}\mathsf{N}\left(\frac{k}{2}, \left(\frac{7}{100}\right)^2\right)$	10^3 10^6	0.92 0.88
8	$\frac{1}{10}\mathsf{N}(0,1) + \frac{9}{10}\mathsf{N}\left(0, \left(\frac{1}{10}\right)^2\right)$	100 10^5	0.63 0.81
9	$\frac{3}{4}\mathsf{N}(0,1) + \frac{1}{4}\mathsf{N}\left(\frac{3}{2}, \left(\frac{1}{3}\right)^2\right)$	10^3 10^6	0.84 0.81
10	Log-Normal with $\mu = 0$ and $\sigma = 1$	100 10^5	0.88 0.81

Table 9.1: Results over 10 independent simulation experiments. In all cases the domain was assumed to be $\mathbb{R}$.

plug-in bandwidth selectors. Our criterion for the comparison is the numerical approximation to

$$R = \frac{||g(\,\cdot\,; \widehat{t^*}) - f||^2}{||\widehat{f}(\,\cdot\,; {}_*\widehat{t}) - f||^2},$$

that is, the ratio of the integrated squared error of the diffusion estimator to the integrated squared error of the Gaussian kernel density estimator.

Table 1 shows the average results over 10 independent trials for a number of different test cases. The second column displays the target density and the third column shows the sample size used for the experiments. In the table $\mathsf{N}(\mu, \sigma^2)$ denotes a Gaussian density with mean μ and variance σ^2. Most test problems are taken from [59]. For each test case we conducted a simulation run with both a relatively small sample size and a relatively large sample size. The table shows that, unlike the standard variable location-scale estimators [60, 75], the diffusion estimator does not require any clipping procedures in order to retain its good performance for large sample sizes.

Finally, we give a two-dimensional density estimation example, which to the best of our knowledge cannot be handled satisfactorily by existing methods [33, 44] due to the boundary bias effects.

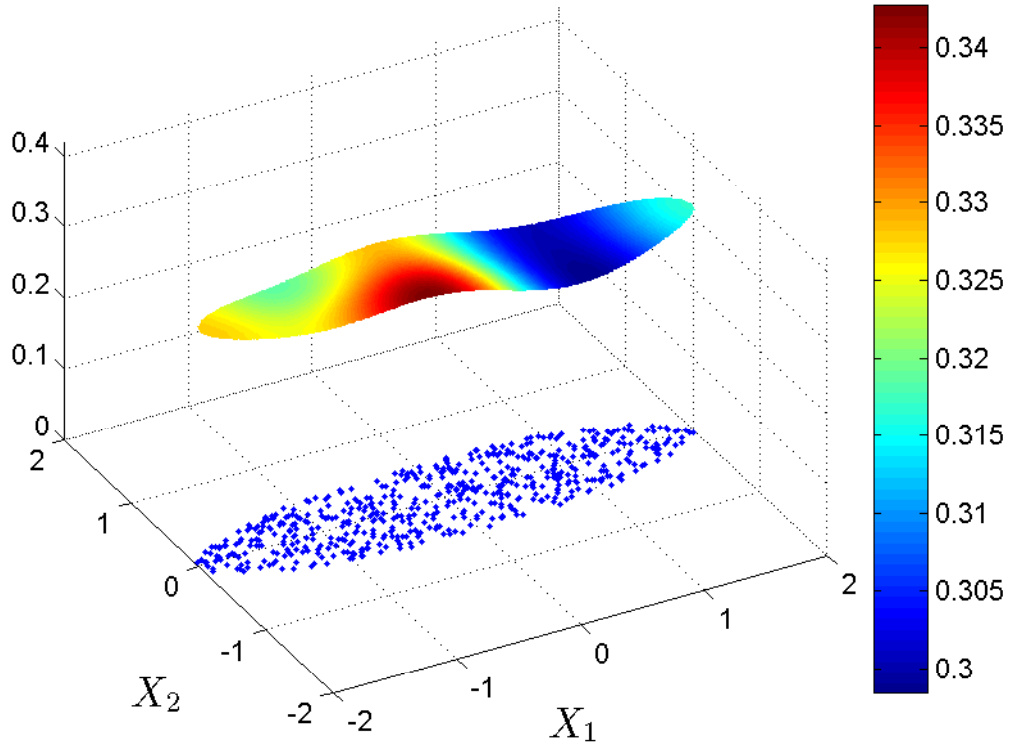

Figure 9.2: A two-dimensional example with 600 points generated uniformly within the ellipse $\mathscr{X} = \{\mathbf{x} : x_1^2 + (4x_2)^2 = 4\}$.

The two-dimensional version of equation (7.2) is:

$$\begin{aligned}
\frac{\partial \widehat{f}}{\partial t}(\mathbf{x};t) &= \frac{1}{2}\left(\frac{\partial^2 \widehat{f}}{\partial x_1^2}(\mathbf{x};t) + \frac{\partial^2 \widehat{f}}{\partial x_2^2}(\mathbf{x};t)\right), \ \forall t > 0, \ \mathbf{x} \in \mathscr{X} \\
\widehat{f}(\mathbf{x};0) &= \Delta(\mathbf{x}) \\
\mathbf{n} \cdot \nabla \widehat{f}(\mathbf{x};t) &= 0, \quad \forall t > 0,
\end{aligned}$$

where $\mathbf{x} = (x_1, x_2)$ belongs to the set $\mathscr{X} \subseteq \mathbb{R}^2$, the initial condition $\Delta(\mathbf{x})$ is the empirical density of the data, and in the Neumann boundary condition $\mathbf{n}$ denotes the unit outward normal to the boundary $\partial\mathscr{X}$ at $\mathbf{x}$. The particular example which we consider is the density estimation of 600 uniformly distributed points on the domain $\mathscr{X} = \{\mathbf{x} : x_1^2 + (4x_2)^2 = 4\}$. We assume that the domain of the data $\mathscr{X}$ is known prior to the estimation. Figure 9.4 shows $\widehat{f}(\mathbf{x};\widehat{t^*})$ on $\mathscr{X} = \{\mathbf{x} : x_1^2 + (4x_2)^2 = 4\}$, that is, it shows the numerical solution of the two-dimensional PDE at time $\widehat{t^*} = 0.13$ on the set $\mathscr{X}$. The bandwidth was determined using the bandwidth selection procedure described in Section 9.2. We emphasize the satisfactory way in which the

pdf $\widehat{f}(\mathbf{x};\widehat{t}^*)$ handles any boundary bias problems. It appears that currently existing methods [33, 36, 44, 45] cannot handle such two-dimensional density estimation problems either because the geometry of the set $\mathscr{X}$ is too complex, or because the resulting estimator is not a bona-fide pdf.

Chapter 10

Diffusion Estimator Synposis

In this chapter we draw conclusions and point to possible future extensions of the diffusion estimator using nonlinear parabolic Partial Differential Equations.

We have presented a new kernel density estimator based on a linear diffusion process. The key idea is to construct an adaptive kernel by considering the most general linear diffusion with its stationary density equal to a pilot density estimate. The resulting diffusion estimator unifies many of the existing ideas about adaptive smoothing. In addition, the estimator is consistent at boundaries. Numerical experiments suggest good practical performance. As future research, the proposed estimator can be extended in a number of ways. First, we can construct kernel density estimators based on Lévy processes, which will have the diffusion estimator as a special case. The kernels constructed via a Lévy process could be tailored for data for which smoothing with the Gaussian kernel density estimator or diffusion estimator is not optimal. Such cases arise when the data is a sample from a heavy-tailed distribution. Second, more subtle and interesting smoothing models can be constructed by considering non-linear parabolic PDEs. One such candidate is the quasilinear parabolic PDE with diffusivity that depends on the density exponentially:

$$\frac{\partial}{\partial t} g(x;t) = \frac{\partial}{\partial x}\left(\mathrm{e}^{-\alpha\, g(x;t)} \frac{\partial}{\partial x} g(x;t) \right), \quad \alpha > 0.$$

Another viable model is the semilinear parabolic PDE:

$$\frac{\partial}{\partial t}\left(\mathrm{e}^{u(x;t)} \right) = \frac{1}{2}\frac{\partial^2}{\partial x^2} u(x;t),$$

where $u(x;t) = \ln(g(x;t))$ is the logarithm of the density estimator. The Cauchy density $\frac{t}{\pi(x^2+t^2)}$ is a particular solution and thus the model could be useful for smoothing heavy-tailed data. All such nonlinear models will provide adaptive smoothing without the need for a pilot run, but at the cost of increased model complexity.

Appendix A

Appendix

In this chapter we introduce the Generalized Splitting method. We explain the context in which the method is applied and how it differs from the classical splitting method.

In this appendix we present the technical details for the proofs of the properties of the diffusion estimator. In addition, we include a description of our plug-in rule in 2 dimensions.

A.1 Gaussian kernel density estimator properties

We use $||\cdot||$ to denote the Euclidean norm on $\mathbb{R}$.

Theorem A.1.1 *Let $t = t_N$ be such that $\lim_{N\to\infty} t_N = 0$ and $\lim_{N\to\infty} N\sqrt{t_N} = \infty$. Assume that f'' is a continuous square-integrable function. The integrated squared bias and integrated variance of the Gaussian kernel density estimator* (7.1) *have asymptotic behavior:*

$$||\mathbb{E}_f[\widehat{f}(\,\cdot\,;t)] - f||^2 = \frac{1}{4}t^2||f''||^2 + o(t^2),\ N\to\infty \tag{A.1}$$

and

$$\int \mathrm{Var}_f[\widehat{f}(x;t)]\,dx = \frac{1}{2N\sqrt{\pi t}} + o((N\sqrt{t})^{-1}),\ N\to\infty, \tag{A.2}$$

respectively. The first-order asymptotic approximation of MISE, denoted AMISE, is thus given by

$$\mathrm{AMISE}\{\widehat{f}\}(t) = \frac{1}{4}t^2||f''||^2 + \frac{1}{2N\sqrt{\pi t}}. \tag{A.3}$$

The asymptotically optimal value of t is the minimizer of the AMISE:

$${}_*t = \left(\frac{1}{2N\sqrt{\pi}\,||f''||^2}\right)^{2/5}, \tag{A.4}$$

giving the minimum value

$$\mathrm{AMISE}\{\widehat{f}\}({}_*t) = N^{-4/5}\,\frac{5||f''||^{2/5}}{4^{7/5}\pi^{2/5}}. \tag{A.5}$$

For a simple proof see [79].

A.2 Proof of Lemma 8.2.1

We seek to establish the behavior of the solution of (8.5) and (8.4) as $t \downarrow 0$. We use the Wentzel-Kramers-Brillouin-Jeffreys (WKBJ) method described in [2, 16, 42, 63]. In the WKBJ method we look for an asymptotic expansion of the form:

$$\kappa(x,y;t) \sim \mathrm{e}^{-\frac{1}{2t}s^2(x,y)} \sum_{m=0}^{\infty} t^{m-1/2} C_m(x,y), \quad t \downarrow 0, \tag{A.6}$$

where $\{C_m(x,y)\}$ and $s(x,y)$ are unknown functions. To determine $s(x,y)$ and $\{C_m(x,y)\}$, we substitute the expansion into (8.4) and, after canceling the exponential term, equate coefficients of like powers of t. This matching of the powers of t leads to solvable ODEs, which determine the unknown functions. Eliminating the leading order $O(t^{-5/2})$ term gives the ODE for s:

$$a(x)\left[\frac{\partial}{\partial x}s(x,y)\right]^2 - p(x) = 0. \tag{A.7}$$

Setting the next highest order $O(t^{-3/2})$ term in the expansion to zero gives the ODE:

$$\begin{aligned} 0 = {} & 2\,a\,(x)\,s\,(x,y)\ \frac{\partial s}{\partial x}\,\frac{dp}{dx}\,p\,(x)\,C_0\,(x,y) - 2\,a\,(x)\,s\,(x,y)\,\frac{\partial s}{\partial x}\,p^2\,(x)\,\frac{\partial C_0}{\partial x} \\ & + p^3\,(x)\,C_0\,(x,y) + s^2\,(x,y)\,p^3\,(x)\,C_1\,(x,y) - \frac{da}{dx}\,p^2\,(x)\,s\,(x,y)\,\frac{\partial s}{\partial x}\,C_0\,(x,y) \\ & + a\,(x)\,s^2\,(x,y)\left(\frac{\partial s}{\partial x}\right)^2 p^2\,(x)\,C_1\,(x,y) - a\,(x)\left(\frac{\partial s}{\partial x}\right)^2 p^2\,(x)\,C_0\,(x,y) \\ & - a\,(x)\,s\,(x,y)\ \frac{\partial^2 s}{\partial x^2}\,p^2\,(x)\,C_0\,(x,y)\,, \end{aligned} \tag{A.8}$$

To determine a unique solution to (A.7) we impose the condition $s(x,x) = 0$, which is necessary, but not sufficient, to ensure that $\lim_{t\downarrow 0}\kappa(x,y;t) = \delta(x-y)$. This gives the solution

$$s(x,y) = \int_y^x \sqrt{\frac{p(s)}{a(s)}}ds.$$

Substituting this solution into (A.8) and simplifying gives an equation without $C_1(x,y)$:

$$C_0(x,y)\,p(x)\,\frac{da}{dx} + 4\,a(x)\,p(x)\,\frac{\partial C_0}{\partial x} - 3\,C_0(x,y)\,\frac{dp}{dx}\,a(x) = 0, \tag{A.9}$$

whence we have the general solution $C_0(x,y) = h(y)\, p^{3/4}(x)\, a^{-1/4}(x)$ for some as yet unknown function of y, $h(y)$. To determine $h(y)$ we require that the kernel $\widetilde{\kappa}(x,y;t)$ satisfies the detailed balance equation (8.9). This ensures that $\widetilde{\kappa}(x,y;t)$ also satisfies (8.5). It follows that $C_0(x,y)$ has to satisfy $p(y)C_0(x,y) = p(x)C_0(y,x)$, which after rearranging gives

$$h(x)(a(x)p(x))^{1/4} = h(y)(a(y)p(y))^{1/4}.$$

A separation of variables argument now gives $h(y)(a(y)p(y))^{1/4} = const.$ and hence

$$C_0(x,y) = const.\ (a(y)p(y))^{-1/4}\, p^{3/4}(x)\, a^{-1/4}(x).$$

We still need to determine the arbitrary constant. The constant is chosen so that

$$\lim_{t\downarrow 0} \int_{-\infty}^{\infty} \widetilde{\kappa}(x,y;t)\, dx = 1,$$

which ensures that $\lim_{t\downarrow 0} \widetilde{\kappa}(x,y;t) = \delta(x-y)$. This final condition yields:

$$C_0(x,y) = \frac{p(x)}{\sqrt{2\pi}(a(y)p(y)a(x)p(x))^{1/4}},$$

and hence

$$\widetilde{\kappa}(x,y;t) = \frac{p(x)}{\sqrt{2\pi t}\ [p(x)a(x)a(y)p(y)]^{1/4}} \exp\left\{ -\frac{1}{2t}\left[\int_y^x \sqrt{\frac{p(s)}{a(s)}}\, ds \right]^2 \right\}.$$

Remark A.2.1 Matching higher powers of t gives first order linear ODEs for the rest of the unknown functions $\{C_m(x,y), m \geqslant 1\}$. The ODE for each $C_m(x,y),\ m = 1,2,3,\ldots$ is:

$$as'(C_m/p)' + \left(\frac{(as')'}{2p} + (m-1/2) \right) C_m = (a(C_{m-1}/p)')', \quad C_m(y,y) = 0,$$

where all derivatives apply to the variable x and y is treated as a constant. Thus, in principle, all functions $\{C_m(x,y)\}$ can be uniquely determined.

It can be shown, see [16], that the expansion (A.6) is valid under the conditions that a, p and all their derivatives are bounded from above, and $p(x) \geqslant p_0 > 0,\ a(x) \geqslant a_0 > 0$. Here we only establish the validity of the leading order approximation $\widetilde{\kappa}$ under the milder conditions (8.11). We do not attempt to prove the validity of the higher order terms in (A.6) under the weaker conditions. The proof of the following lemma uses arguments similar to the ones given in [16].

Lemma A.2.1 *Let $a(x)$ and $p(x)$ satisfy conditions (8.11). Then, for all $t \in (0, t_0]$, where $t_0 > 0$ is some constant independent of x and y, there holds:*

$$|\kappa(x, y; t) - \widetilde{\kappa}(x, y; t)| \leqslant const.\, C_0(x, y)\, t^{1/4} e^{-\frac{s^2(x,y)}{2t}},\ \forall x, y.$$

To prove the lemma we first begin by proving the following auxiliary results.

Proposition A.2.1 *Define*

$$\ell(z) = \ell(z; x, y, t, \tau) = \frac{s^2(x, z)}{2(t-\tau)} + \frac{s^2(z, y)}{2\tau}.$$

Then for $\tau \in (0, t)$ we have

$$\ell(z) \geqslant \frac{s^2(x, y)}{2t}.$$

Moreover, there exists a unique $z_0 = z_0(x, y, t, \tau)$ for which $\ell(z_0) = \frac{s^2(x,y)}{2t}$, and $\ell(z)$ is increasing for $z > z_0$ and decreasing for $z < z_0$.

Proof: We have

$$\ell(z) = \frac{1}{2(t-\tau)} \left(\int_z^x \sigma^{-1}(s)\, ds \right)^2 + \frac{1}{2\tau} \left(\int_y^z \sigma^{-1}(s)\, ds \right)^2,$$

and hence

$$\ell'(z) = \frac{-\sigma^{-1}(z)}{t-\tau} \int_z^x \sigma^{-1}(s)\, ds + \frac{\sigma^{-1}(z)}{\tau} \int_y^z \sigma^{-1}(s)\, ds. \tag{A.10}$$

For $x \neq y$, $\ell'(y) > 0, \ell'(x) < 0$, and therefore by the continuity of ℓ', there exists $z_0 \in (x, y)$: $\ell'(z_0) = 0$. For $x = y$, set $z_0 = x$. Setting $z = z_0$ in (A.10),

$$\frac{1}{t-\tau} \int_{z_0}^x \sigma^{-1}(s)\, ds = \frac{1}{\tau} \int_y^{z_0} \sigma^{-1}(s)\, ds. \tag{A.11}$$

Therefore, $\int_{z_0}^x \sigma^{-1}(s)\, ds = \frac{t-\tau}{\tau} \int_y^{z_0} \sigma^{-1}(s)\, ds$ and adding $\int_y^{z_0} \sigma^{-1}(s)\, ds$ to both sides we obtain

$$\int_y^x \sigma^{-1}(s)\, ds = \frac{t}{\tau} \int_y^{z_0} \sigma^{-1}(s)\, ds,$$

from which we see that (A.11) is also equal to $\frac{1}{t} \int_y^x \sigma^{-1}(s)\, ds$. Hence, by substitution $\ell(z_0) = \frac{1}{2t} \left(\int_y^x \sigma^{-1}(s)\, ds \right)^2$, as required. Finally, note that if $F(z) = \ell(z) - \frac{t}{2\tau(t-\tau)} \left(\int_{z_0}^z \sigma^{-1}(s)\, ds \right)^2$, then $F'(z) = 0$ for all z. Hence, $F(z) = F(z_0) = \ell(z_0)$ and

$$\ell(z) = \ell(z_0) + \frac{t}{2\tau(t-\tau)} \left(\int_{z_0}^z \sigma^{-1}(s)\, ds \right)^2. \tag{A.12}$$

As a consequence of Proposition A.2.1, we have the following result.

Proposition A.2.2 *Assuming* $\lim_{z\to\pm\infty}\int_{z_0}^{z}\sigma^{-1}(s)ds=\pm\infty$, *we have the following equality:*

$$\int_0^t\sqrt{\int_{-\infty}^{\infty}\left(\frac{e^{-\frac{s^2(x,z)}{2(t-\tau)}}}{\sqrt{t-\tau}}\frac{e^{-\frac{s^2(z,y)}{2\tau}}}{\sqrt{\sigma(z)}\sqrt{\tau}}\right)^2 dz\,d\tau}=2\pi^{-1/4}t^{1/4}\,\Gamma^2(3/4)\,\mathrm{e}^{-\frac{s^2(x,y)}{2t}},$$

$$=c_2\,t^{1/4}\mathrm{e}^{-\frac{s^2(x,y)}{2t}},$$

where c_2 *is a constant (indeed* $c_2=2\pi^{-1/4}\Gamma^2(3/4)$*).*

Proof: We have

$$\int_{-\infty}^{\infty}\frac{\mathrm{e}^{-\frac{s^2(x,z)}{(t-\tau)}}}{t-\tau}\frac{\mathrm{e}^{-\frac{s^2(z,y)}{\tau}}}{\sigma(z)\tau}dz=\frac{1}{(t-\tau)\tau}\mathrm{e}^{-2\ell(z_0)}\int_{-\infty}^{\infty}\sigma^{-1}(z)\mathrm{e}^{-\frac{\left(\int_{z_0}^{z}\sigma^{-1}(s)ds\right)^2}{\tau(t-\tau)/t}}dz$$

$$=\frac{1}{\sqrt{t(t-\tau)\tau}}\mathrm{e}^{-2\ell(z_0)}\int_{-\infty}^{\infty}\mathrm{e}^{-v^2}dv,$$

with the change of variable $v(z)=\frac{1}{\sqrt{\tau(t-\tau)/t}}\int_{z_0}^{z}\sigma^{-1}(s)\,ds$. Then, the result follows from the fact that $\int_0^t(\tau(t-\tau))^{-1/4}d\tau=2\pi^{-1/2}t^{1/2}\Gamma^2(3/4)$.

Given these two auxiliary results, we proceed with the proof of Lemma A.2.1. Writing

$$\kappa^*(x,y;t)=\frac{\partial}{\partial t}\widetilde{\kappa}(x,y;t)-L\widetilde{\kappa}(x,y;t)=-\frac{\mathrm{e}^{-\frac{s^2(x,y)}{2t}}}{\sqrt{t}}LC_0(x,y),$$

we define inductively the following sequence of function $\{\varrho_j\}$, starting with $\varrho_0=0$:

$$\varrho_{j+1}(x,y;t)=-\kappa^*(x,y;t)-\int_0^t\int_{-\infty}^{\infty}\kappa^*(x,z;t-\tau)\varrho_j(z,y;\tau)dzd\tau,\quad j=1,2,\ldots.$$

Note in particular that $\varrho_1=-\kappa^*$. We will show that there exists a limit of $\{\varrho_j\}$. We begin by proving via induction that for $j\geqslant 1$, $x,y\in\mathbb{R}$, $t\in(0,t_0]$, where

$$t_0=\min\left\{\left(\frac{\sqrt{2\pi}}{2c_1c_2}\right)^{4/3},1\right\},$$

there holds:

$$|\varrho_{j+1}(x,y,t)-\varrho_j(x,y,t)|\leqslant\frac{c_3}{2^j}|LC_0(x,y)|t^{1/4}\mathrm{e}^{-\frac{s^2(x,y)}{2t}},\tag{A.13}$$

where $c_3 = 2c_1c_2/\sqrt{2\pi}$. Firstly, we calculate for $j = 1$:

$$\varrho_2(x,y,t) = -\kappa^*(x,y,t) + \int_0^t \int_{-\infty}^{\infty} \kappa^*(x,z,t-\tau)\kappa^*(z,y,\tau)dzd\tau.$$

Therefore, we have the following bound:

$$\begin{aligned}
|\varrho_2(x,y,t) - \varrho_1(x,y,t)| &\leqslant \int_0^t \int_{-\infty}^{\infty} |\kappa^*(x,z,t-\tau)\kappa^*(z,y,\tau)|dzd\tau \\
&= \int_0^t \int_{-\infty}^{\infty} \frac{\mathrm{e}^{-\frac{s^2(x,z)}{2(t-\tau)}}}{\sqrt{t-\tau}} \frac{\mathrm{e}^{-\frac{s^2(z,y)}{2\tau}}}{\sqrt{\tau}} |LC_0(x,z)LC_0(z,y)|dzd\tau \\
&= \int_0^t \int_{-\infty}^{\infty} \frac{\mathrm{e}^{-\frac{s^2(x,z)}{2(t-\tau)}}}{\sqrt{t-\tau}} \frac{\mathrm{e}^{-\frac{s^2(z,y)}{2\tau}}}{\sqrt{\sigma(z)\tau}} \sqrt{\sigma(z)} |LC_0(x,y)| \frac{|Lq(z)|}{\sqrt{2\pi}(a(z)p(z))^{1/4}} dzd\tau \\
&= \frac{1}{\sqrt{2\pi}} |LC_0(x,y)| \int_0^t \int_{-\infty}^{\infty} \frac{\mathrm{e}^{-\frac{s^2(x,z)}{2(t-\tau)}}}{\sqrt{t-\tau}} \frac{\mathrm{e}^{-\frac{s^2(z,y)}{2\tau}}}{\sqrt{\sigma(z)\tau}} \frac{|Lq(z)|}{q(z)} dzd\tau \\
&\leqslant \frac{1}{\sqrt{2\pi}} |LC_0(x,y)|\, c_1\, c_2\, t^{1/4}\, \mathrm{e}^{-\frac{s^2(x,y)}{2t}},
\end{aligned}$$

where the last inequality follows from the Cauchy-Schwartz inequality, Proposition A.2.2, and assumption (8.11). We thus have

$$|\varrho_2(x,y,t) - \varrho_1(x,y,t)| \leqslant \frac{c_3}{2} |LC_0(x,y)| t^{1/4} \mathrm{e}^{-\frac{s^2(x,y)}{2t}}.$$

Next, assume the induction statement is true for $2, 3, \ldots, j-1$. Then,

$$\begin{aligned}
|\varrho_{j+1}(x,y,t) - \varrho_j(x,y,t)| &\leqslant \int_0^t \int_{-\infty}^{\infty} |\kappa^*(x,z,t-\tau)||\varrho_j(z,y,\tau) - \varrho_{j-1}(z,y,\tau)|dzd\tau \\
&\leqslant \int_0^t \int_{-\infty}^{\infty} \frac{\mathrm{e}^{-\frac{s^2(x,z)}{2(t-\tau)}}}{\sqrt{t-\tau}} |LC_0(x,z)| \frac{c_3}{2^{j-1}} |LC_0(z,y)| \tau^{1/4} \mathrm{e}^{-\frac{s^2(z,y)}{2\tau}} dzd\tau \\
&\leqslant \frac{c_3}{2^{j-1}} |LC_0(x,y)| \int_0^t \int_{-\infty}^{\infty} \frac{\mathrm{e}^{-\frac{s^2(x,z)}{2(t-\tau)}}}{\sqrt{t-\tau}} \frac{\mathrm{e}^{-\frac{s^2(z,y)}{2\tau}}}{\sqrt{\sigma(z)\tau}} \tau^{3/4} \frac{|Lq(z)|}{\sqrt{2\pi}q(z)} dzd\tau \\
&\leqslant \frac{c_3}{2^{j-1}} |LC_0(x,y)|\, t^{1/4}\, \mathrm{e}^{-\frac{s^2(x,y)}{2t}} t_0^{3/4} \frac{c_1\, c_2}{\sqrt{2\pi}}
\end{aligned}$$

The last line follows from the Cauchy-Schwartz inequality and the fact that $\tau^{3/4} \leqslant t^{3/4} \leqslant t_0^{3/4}$. Since $t_0^{3/4} \frac{c_1 c_2}{\sqrt{2\pi}} \leqslant \frac{1}{2}$, we obtain:

$$|\varrho_{j+1}(x,y,t) - \varrho_j(x,y,t)| \leqslant \frac{c_3}{2^j} |LC_0(x,y)| t^{1/4} \mathrm{e}^{-\frac{s^2(x,y)}{2t}}.$$

This establishes (A.13). Next, we have the bound for all $j \geqslant 1$:

$$\begin{aligned} |\varrho_j(x,y,t)| &\leqslant |\varrho_1(x,y,t)| + \sum_{j=1}^{\infty} \frac{c_3}{2^j} |LC_0(x,y)| t^{1/4} \mathrm{e}^{-\frac{s^2(x,y)}{2t}} \\ &\leqslant |LC_0(x,y)| \left(\frac{1}{\sqrt{t}} + c_3 t^{1/4} \right) \mathrm{e}^{-\frac{s^2(x,y)}{2t}} \\ &\leqslant |LC_0(x,y)| \frac{2}{\sqrt{t}} \mathrm{e}^{-\frac{s^2(x,y)}{2t}}. \end{aligned} \tag{A.14}$$

In the light of (A.14) and (A.13), the pointwise limit

$$\varrho(x,y,t) = \lim_{j \to \infty} \varrho_j(x,y,t)$$

exists on $\mathbb{R} \times \mathbb{R} \times (0, t_0)$. In addition, $\varrho(x,y,t)$ satisfies the limiting equation:

$$0 = \kappa^*(x,y,t) + \varrho(x,y,t) + \int_0^t \int_{-\infty}^{\infty} \kappa^*(x,z,t-\tau) \varrho(z,y,\tau) dz d\tau,$$

and indeed

$$\kappa(x,y;t) - \widetilde{\kappa}(x,y;t) = \int_0^t \int_{-\infty}^{\infty} \widetilde{\kappa}(x,z,t-\tau) \varrho(z,y,\tau)\, dz d\tau\,. \tag{A.15}$$

In order to see this, we can apply directly the arguments of Section 5 of [16] in the case $N = 0$, see also Section 1.3 of [21]. Hence, we can take the limit in (A.14) to conclude

$$|\varrho(x,y,t)| \leqslant 2 |LC_0(x,y)| t^{-1/2} \mathrm{e}^{-\frac{s^2(x,y)}{2t}} \tag{A.16}$$

for $t \in (0, t_0]$. The claim of the lemma then follows from

$$\begin{aligned} |\kappa(x,y;t) - \widetilde{\kappa}(x,y;t)| &\leqslant \int_0^t \int_{-\infty}^{\infty} \widetilde{\kappa}(x,z,t-\tau) |\varrho(z,y,\tau)|\, dz d\tau \\ &\leqslant 2 \int_0^t \int_{-\infty}^{\infty} \frac{\mathrm{e}^{-\frac{s^2(x,z)}{2(t-\tau)}}}{\sqrt{t-\tau}} C_0(x,z) \frac{\mathrm{e}^{-\frac{s^2(z,y)}{2\tau}}}{\sqrt{\tau}} |LC_0(z,y)| dz d\tau \\ &\leqslant \frac{2}{\sqrt{2\pi}} C_0(x,y) \int_0^t \int_{-\infty}^{\infty} \frac{\mathrm{e}^{-\frac{s^2(x,z)}{2(t-\tau)}}}{\sqrt{t-\tau}} \frac{\mathrm{e}^{-\frac{s^2(z,y)}{2\tau}}}{\sqrt{\sigma(z)\tau}} \frac{|Lq(z)|}{q(z)} dz d\tau \\ &\leqslant 2 C_0(x,y) t^{1/4} \mathrm{e}^{-\frac{s^2(x,y)}{2t}} \frac{c_1 c_2}{\sqrt{2\pi}} = c_3 C_0(x,y) t^{1/4} \mathrm{e}^{-\frac{s^2(x,y)}{2t}}. \end{aligned}$$

A.3 Proof of Theorem 8.2.1

Note that (8.12) is given by $\int_{-\infty}^{\infty} \kappa(x,y;t)f(y)dy - f(x)$, and from (8.5) we have

$$\begin{aligned}\frac{\partial}{\partial t}g(x;t) &= \int_{\mathscr{X}} f(y)L^*\kappa(x,y;t)\,dy \\ &= -\frac{1}{2}\frac{d}{dy}\left(\frac{f(y)}{p(y)}\right)a(y)\kappa(x,y;t)\Big|_{y\in\partial\mathscr{X}} + \int_{\mathscr{X}} \kappa(y,x;t)Lf(x)dx.\end{aligned}$$

Given that $\mathscr{X} \equiv \mathbb{R}$, Lemma 8.2.1 gives $\kappa(x,y;t)\Big|_{y\in\partial\mathscr{X}} \sim \widetilde{\kappa}(x,y;t)\Big|_{y=-\infty}^{y=\infty}$, $t \downarrow 0$. The last term is zero since for fixed x,

$$\lim_{y\to\pm\infty}\left[\int_y^x \sqrt{\frac{p(s)}{a(s)}}\,ds\right]^2 = \infty,$$

and hence $\lim_{y\to\pm\infty}\widetilde{\kappa}(x,y;t) = 0$. We have,

$$g(x;t) = g(x;0) + t\frac{\partial}{\partial t}g(x;t)\Big|_{t=0} + O(t^2),$$

because $g(x;t), t > 0$ is smooth (see, e.g., Theorem IV $\cdot$ 10 $\cdot$ 1 in [48]). Therefore,

$$g(x;t) = f(x) + t\,Lf(x) + O(t^2),$$

and (8.12) and (8.13) follow. We now proceed to demonstrate (8.14). First, the second moment has the behavior

$$\begin{aligned}\mathbb{E}_f[\kappa^2(x,Y;t)] &= \int_{\mathscr{X}} f(y)\kappa^2(x,y;t)dy \\ &\sim \int_{\mathscr{X}} f(y)\widetilde{\kappa}^2(x,y;t)dy \\ &\sim \frac{p^2(x)}{2\pi t\sqrt{p(x)a(x)}}\int_{-\infty}^{\infty}\frac{f(y)}{\sqrt{p(y)a(y)}}\,\mathrm{e}^{-\frac{1}{2}\left[\sqrt{\frac{2}{t}}\int_x^y\sqrt{\frac{p(s)}{a(s)}}\,ds\right]^2}dy.\end{aligned}$$

We can simplify the last expression by the change of variable $u = \sqrt{\frac{2}{t}}\int_x^y \sqrt{\frac{p(s)}{a(s)}}\,ds$. This gives

$$\frac{p^2(x)}{2\pi\sqrt{2t}\sqrt{p(x)a(x)}}\int_{-\infty}^{\infty}\frac{f(y(u,t))}{p(y(u,t))}\,\mathrm{e}^{-\frac{u^2}{2}}du,$$

where $y(u,t) = y(u,0) + \sqrt{t}\frac{\partial y}{\partial\sqrt{t}}\Big|_{t=0} + O(t) = x + u\sqrt{\frac{ta(x)}{2p(x)}} + O(t)$ is a Taylor expansion of $y(u,t)$

at $\sqrt{t}=0$. Therefore, $\frac{f(y(u,t))}{p(y(u,t))} \sim \frac{f(x)}{p(x)}$ as $t \downarrow 0$, and

$$\frac{p^2(x)}{2\pi\sqrt{2t}\sqrt{p(x)a(x)}} \int_{-\infty}^{\infty} \frac{f(y(u,t))}{p(y(u,t))} \mathrm{e}^{-\frac{u^2}{2}} du \sim \frac{1}{2\sqrt{\pi t}} f(x) \sqrt{\frac{p(x)}{a(x)}}, \quad t \downarrow 0.$$

Hence, from (8.3) have

$$\mathrm{Var}_f[g(x;t)] = \frac{1}{N}\mathbb{E}_f[\kappa^2(x,Y;t)] - \frac{1}{N}\mathbb{E}_f[\kappa(x,Y;t)]^2 \sim \frac{f(x)}{2N\sqrt{\pi t}\,\sigma(x)}, \quad t \downarrow 0,$$

from which (8.15) and (8.14) follow.

A.4 Consistency at Boundary

As in [79], we consider the case where the support of f is $[0,\infty]$. The consistency of the estimator near $x=0$ is analyzed by considering the pointwise bias of estimator (8.3) at a point x_N such that x_N is $O(\sqrt{t_N})$ away from the boundary, that is, x_N is approaching the boundary at the same rate at which the bandwidth is approaching 0. We then have the following result, which shows that the diffusion estimator (8.3), and hence its special case (7.3), is consistent at the boundaries.

Proposition A.4.1 *Let $\mathscr{X} \equiv [0,\infty]$, and assume that $x = x_N = \alpha\sqrt{t_N}$ for some constant $\alpha \in [0,1]$, where $\lim_{N\to\infty} t_N = 0$ and $\lim_{N\to\infty} N\sqrt{t_N} = \infty$. Then for the diffusion estimator (8.3) we have*

$$\mathbb{E}_f g(x_N;t) = f(x_N) + O(\sqrt{t_N}), \quad N \to \infty.$$

Hence, the diffusion estimator (8.3) is consistent at the boundaries.

Proof: First, we differentiate both sides of $\mathbb{E}_f g(x;t) = \int_0^1 f(y)\kappa(x;y;t)\,dy$ with respect to t and use (8.5) to obtain:

$$\begin{aligned}
\frac{\partial}{\partial t}\mathbb{E}_f g(x;t) &= \int_0^\infty f(y)\frac{\partial}{\partial t}\kappa(x;y;t)\,dy \\
&= \int_0^\infty f(y) L^*\kappa(x;y;t) dy \\
&= -\frac{1}{2}\left(\frac{f(y)}{p(y)}\right)' a(y)\ \kappa(x;y;t)\Big|_{y=0}^{y=\infty} + \int_0^\infty \kappa(x;y;t) L f(y)\,dy.
\end{aligned}$$

Second, we show that $\kappa(\alpha\sqrt{t_N};0;t_N) = O(t^{-1/2})$ and $\lim_{y\to\infty}\kappa(\alpha\sqrt{t_N};y;t_N) = o(1)$, and $\int_0^1 \kappa(x;y;t_N) L f(y)\,dy = O(1)$ as $N \to \infty$. To this end we consider the small bandwidth

behavior of κ. It is easy to verify using Lemma 8.2.1 that the *boundary kernel*:

$$\kappa_B(x,y;t) = \widetilde{\kappa}(x,y;t) + \widetilde{\kappa}(x,-y;t)$$

satisfies

$$\frac{\partial}{\partial t}\kappa_B(x,y;t) = L^*\kappa_B(x,y;t) + O\left(\mathrm{e}^{-\frac{s^2(x,y)}{2t}}\ t^{-1/2}\right), \quad t \downarrow 0$$

on $x, y \in \mathbb{R}$ with initial condition $\kappa_B(x,y;0) = \delta(x-y)$. In addition, the boundary kernel satisfies the condition $\frac{\partial}{\partial y}\kappa_B(x,y;t)\big|_{y=0} = 0$, and therefore κ_B describes the small bandwidth asymptotics of the solution of the PDE (8.5) on the domain $x, y \in [0,\infty)$ with boundary condition $\frac{\partial}{\partial y}\kappa(x,y;t)\big|_{y=0} = 0$. Hence, we have

$$\kappa(\alpha\sqrt{t};0;t) \sim \kappa_B(\alpha\sqrt{t};0;t) = const.\ t^{-1/2}\mathrm{e}^{O(\sqrt{t})}, \quad t \downarrow 0,$$

and

$$\lim_{y\to\infty} \kappa_B(\alpha\sqrt{t};y;t) = 0, \quad t > 0.$$

Therefore,

$$\frac{\partial}{\partial t}\mathbb{E}_f g(x_N;t_N) = o(1) - O(t_N^{-1/2}), \quad N \to \infty,$$

or

$$\frac{\mathbb{E}_f g(x_N;t_N) - \mathbb{E}_f g(x_N;0)}{t_N} + O(t_N) = O(t_N^{-1/2}) + O(1), \quad N \to \infty,$$

which, after rearranging, gives

$$\mathbb{E}_f g(x_N;t_N) = f(x_N) + O(\sqrt{t_N}), \quad N \to \infty.$$

References

[1] I.S. Abramson. On bandwidth variation in kernel estimates—a square root law. *Ann. Stat.*, 10:1217–1223, 1982.

[2] R. Azencott. Density of diffusions in small time: asymptotic expansions. *Lecture Notes in Mathematics*, 1059:402–498, 1984.

[3] Richard Bellman. *A Brief Introduction to Theta Functions*. Holt, Rinehart and Winston, New York, 1961.

[4] Z. I. Botev. http://www.mathworks.us/matlabcentral/fileexchange/authors/27236. *Kernel density estimation using Matlab*, 2007.

[5] Z. I. Botev. Nonparametric density estimation via diffusion mixing. Technical report, http://espace.library.uq.edu.au. *Department of Mathematics, The University of Queensland*, 2007.

[6] Z. I. Botev. An algorithm for rare-event probability estimation using the product rule of probability theory. *PhD progress report, Department of Mathematics, The University of Queensland, http://espace.library.uq.edu.au/view/UQ:151299*, 2008.

[7] Z. I. Botev and D.P. Kroese. An efficient algorithm for rare-event probability estimation, combinatorial optimization, and counting. *Methodology and Computing in Applied Probability*, 10(4), 2008.

[8] Z. I. Botev and D.P. Kroese. The generalized cross entropy method, with applications to probability density estimation. *Methodology and Computing in Applied Probability*, to appear, 2009.

[9] S. P. Brooks, P. Dellaportas, and G. O. Roberts. An approach to diagnosing total variation convergence of MCMC algorithms. *Journal of Computational and Graphical Statistics*, 1, 1997.

[10] S. P. Brooks and P. Giudici. Markov Chain Monte Carlo convergence assessment via two-way analysis of variance. *Journal of Computational and Graphical Statistics*, 9:266–285, 2000.

[11] S. P. Brooks and G. O. Roberts. Convergence assessment techniques for Markov Chain Monte Carlo. *Statistics and Computing*, 8:319–335, 1998.

[12] F. Cerou and A. Guyader. Adaptive multilevel splitting for rare event analysis. *Stoch. Anal. Appl.*, 25:417–443, 2007.

[13] F. Cerou, P. Del moral, T. Furon, and A. Guyader. Rare-event simulation for a static distribution. *INRIA-00350762*, 2009.

[14] Chaudhuri and Marron. Scale space view of of curve estimation. *The Annals of Statistics*, 2000.

[15] E. Choi and P. Hall. Data sharpening as a prelude to density estimation. *Biometrika*, 86:941–947, 1999.

[16] J. K. Cohen, F. G. Hagin, and J. B. Keller. Short time asymptotic expansions of solutions of parabolic equations. *Journal of Mathematical Analysis and Applications*, 38:82–91, 1972.

[17] I. Csiszár. A class of measures of informativity of observation channels. *Periodic Math. Hungarica*, 2:191–213, 1972.

[18] Luc Devroye. Universal smoothing factor selection in density estimation: theory and practice. *Sociedad de stadistica e Investigacian Operativa*, 6(2), 1997.

[19] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York, 2001.

[20] S. N. Ethier and T. G. Kurtz. *Markov processes. Characterization and Convergence*. Wiley Series In Probability and Mathematical Statistics, 2009.

[21] A. Friedman. *Partial Differential Equations of Parabolic Type*. Prentice-Hall, 1964.

[22] W. Gander and W. Gautschi. Adaptive quadrature - revisited. *BIT Numerical Mathematics*, 40:84–101, 2000.

[23] M. J. J. Garvels. *The splitting method in rare event simulation*. PhD thesis, University of Twente, The Netherlands, October 2000.

[24] M. J. J. Garvels and D. P. Kroese. A comparison of RESTART implementations. In *Proceedings of the 1998 Winter Simulation Conference*, pages 601–609, Washington, DC, 1998.

[25] M. J. J. Garvels, D. P. Kroese, and J. C. W. van Ommeren. On the importance function in RESTART simulation. *European Transactions on Telecommunications*, 13(4), 2002.

[26] A. Gelman and D. Rubin. Inference from iterative simulation using multiple sequences (with discussion). *Statistical Science*, 7:457–511, 1992.

[27] P. Glasserman, P. Heidelberger, P. Shahabuddin, and T. Zajic. A look at multilevel splitting. In H. Niederreiter, editor, *Monte Carlo and Quasi Monte Carlo Methods 1996, Lecture Notes in Statistics*, volume 127, pages 99–108. Springer-Verlag, New York, 1996.

[28] P. Glasserman, P. Heidelberger, P. Shahabuddin, and T. Zajic. A large deviations perspective on the efficiency of multilevel splitting. *IEEE Transactions on Automatic Control*, 43(12):1666–1679, 1998.

[29] J. Gu, P. W. Purdom, J. Franco, and B. W. Wah. Algorithms for the satisfiability (SAT) problem: A survey. *Satisfiability Problem: Theory and Applications*, volume 35 of DIMACS Series in Discrete Mathematics. American Mathematical Society, 1996.

[30] P. Hall. On the bias of variable bandwidth curve estimators. *Biometrika*, 77:523–535, 1990.

[31] P. Hall, T. C. Hu, and J. S. Marron. Improved variable window kernel estimates of probability densities. *Annals of Ststistics*, 23:1–10, 1995.

[32] P. Hall and M. C. Minnotte. High order data sharpening for density estimation. *Journal of the Royal Statistical Society Series B-Statistical Methodology*, 64:141–157, 2002.

[33] P. Hall and B. U. Park. New methods for bias correction at endpoints and boundaries. *Annals of Statistics*, 30:1460–1479, 2002.

[34] J. H. Havrda and F. Charvat. Quantification methods of classification processes:concepts of structural α entropy. *Kybernatica*, 3:30–35, 1967.

[35] M. Hollander and D. A. Wolfe. *Nonparametric Statistical Methods*. John Wiley& Sons, New York, 1973.

[36] M. C. Jones and P. J. Foster. A simple nonnegative boundary correction method for kernel density estimation. *Statistica Sinica*, 6:1005–1013, 1996.

[37] M. C. Jones, J. S. Marron, and S. J. Sheather. Simple boundary correction for kernel density estimation. *Statistical Computing*, 3:135–46, 1993.

[38] M. C. Jones, J. S. Marron, and S. J. Sheather. A brief survey of bandwidth selection for density estimation. *Journal of the American Statistical Association*, 91:401–407, 1996.

[39] M. C. Jones, J. S. Marron, and S. J. Sheather. Progress in data-based bandwidth selection for kernel density estimation. *Computational Statistics*, 11:337–381, 1996.

[40] M. C. Jones and D. F. Signorini. A comparison of higher-order bias kernel density estimators. *Journal of the American Statistical Association*, 92:1063–1073, 1997.

[41] H. Kahn and T. E. Harris. *Estimation of Particle Transmission by Random Sampling.* National Bureau of Standards Applied Mathematics Series, 1951.

[42] Y. Kannai. Off diagonal short time asymptotics for fundamental solutions of diffusion equations. *Comm. in Partial Differential Equations*, 2:781–830, 1977.

[43] J. N. Kapur and H. K. Kesavan. *Generalized Maximum Entropy Principle (With applications).* Standford Educational Press, University of Waterloo, Waterloo, Ontario, Canada, 1987.

[44] R. J. Karunamuni and T. Alberts. A generalized reflection method of boundary correction in kernel density estimation. *Canadian Journal of Statistic*, 33:497–509, 2005.

[45] R. J. Karunamuni and S. Zhang. Some improvements on a boundary corrected kernel density estimator. *Statistics & Probability Letters*, 78:499–507, 2008.

[46] P. E. Kloeden and E. Platen. *Numerical Solution of Stochastic Differential Equations.* Springer, Berlin, 1999.

[47] S. C. Kou, Quing Zhou, and Wing Hung Wong. Equi-energy sampler with applications in statistical inference and statistical mechanics. *The Annals of Statistics*, 34:1581–1619, 2006.

[48] O. A. Ladyženskaja, V. A. Solonnikov, and N. N. Ural'ceva. *Linear and quasilinear equations of parabolic type.* Translated from the Russian by S. Smith. Translations of Mathematical Monographs, Vol. 23 American Mathematical Society, Providence, R.I. 1967 xi+648 pp.

[49] Stig Larsson and Vidar Thomee. *Partial Differential Equations with Numerical Methods.* Springer, 2003.

[50] P. L'Ecuyer, V. Demeres, and B. Tuffin. Splitting for rare-event simulation. *Proceedings of the 2006 Winter Simulation Conference*, 2006.

[51] P. L'Ecuyer, V. Demeres, and B. Tuffin. Rare events, splitting, and quasi-monte carlo. *ACM Transactions on modeling and computer simulation*, 17(2), 2007.

[52] E. L. Lehmann. Model specification: The views of fisher and neyman, and later developments. *Statistical Science*, 5:160–168, 1990.

[53] F. Liang and W. H. Wong. Real-parameter evolutionary monte carlo with applications to bayesian mixture models. *Journal of the American Statistical Association*, 96:653–666, 2001.

[54] J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer-Verlag, New York, 2001.

[55] D. O. Loftsgaarden and C. P. Quesenberry. A nonparametric estimate of a multivariate density function. *Annals of Mathematical Statistics*, 36:1046–59, 1965.

[56] J. S. Marron M. C. Jones and B. U. Park. A simple root n bandwidth selector. *The Annals of Statistics*, 19 4:1919–1932, 1991.

[57] J. S. Marron. An asymptotically efficient solution to the bandwidth problem of kernel density estimation. *The Annals of Statistics*, 13:1011–1023, 1985.

[58] J. S. Marron and D. Ruppert. Transformations to reduce boundary bias in kernel density-estimation. *Journal of the Royal Statistical Society Series B-Methodological*, 56:653–671, 1996.

[59] J. S. Marron and M. P. Wand. Exact mean integrated error. *The Annals of Statistics*, 20:712–736, 1992.

[60] I.J. McKay M.C. Jones and T.C.Hu. Variable location and scale kernel density estimation. *Annals of the Institute of Statistical Mathematics*, 46:521–535, 1994.

[61] G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. John Wiley & Sons, New York, 1997.

[62] G. J. McLachlan and D. Peel. *Finite Mixture Models*. John Wiley & Sons, New York, 2000.

[63] S. A. Molchanov. Diffusion process and Riemannian geometry. *Russian Math. Surveys*, 30:1–63, 1975.

[64] P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo Samplers. *J. Royal Statistical Society B*, 68(3):411–436, 2006.

[65] B. U. Park, S. O. Jeong, and M. C. Jones. Adaptive variable location kernel density estimators with good performance at boundaries. *Journal of Nonparametric Statistics*, 15:61–75, 2003.

[66] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer-Verlag, New York, Second Edition, 2004.

[67] R. Y. Rubinstein and D. P. Kroese. *The Cross-Entropy Method*. Springer-Verlag, New York, 2004.

[68] R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo Method. Second edition.* John Wiley & Sons, New York, 2007.

[69] R. Y. Rubinstein and A. Shapiro. *Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization via the Score Function Method*. John Wiley & Sons, New York, 1993.

[70] M. Samiuddin and G. M. El-Sayyad. On nonparametric kernel density estimates. *Biometrika*, 77(4):865–874, 1990.

[71] D. W. Scott. *Multivariate Density Estimation. Theory, Practice and Visualization*. John Wiley & Sons, 1992.

[72] S. J. Sheather and M. C. Jones. A reliable data-based bandwidth selection method for kernel density estimation. *J.R.Statist.Soc.B*, 53:683–690, 1991.

[73] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.

[74] J. S. Simonoff. *Smoothing Methods in Statistics*. Springer, 1996.

[75] G. R. Terrell and David W. Scott. Variable kernel density estimation. *The Annals of Statistics*, 20:1236–1265, 1992.

[76] M. Villén-Altamirano and J. Villén-Altamirano. RESTART: A method for accelerating rare event simulations. In J. W. Cohen and C. D. Pack, editors, *Proceedings of the 13th International Teletraffic Congress, Queueing, Performance and Control in ATM*, pages 71–76, 1991.

[77] M. Villén-Altamirano and J. Villén-Altamirano. RESTART: A straightforward method for fast simulation of rare events. In J. D. Tew, S. Manivannan, D. A. Sadowski, and A.F. Seila, editors, *Proceedings of the 1994 Winter Simulation Conference*, pages 282–289, 1994.

[78] M. Villén-Altamirano and J. Villén-Altamirano. About the efficiency of RESTART. In *Proceedings of the RESIM'99 Workshop*, pages 99–128. University of Twente, the Netherlands, 1999.

[79] M. P. Wand and M. C. Jones. *Kernel Smoothing.* Chapman & Hall, 1995.

[80] D. J. A. Welsh. *Complexity: Knots, Coloring and Counting.* Cambridge University Press, Cambridge, 1993.